コンピュータ・グラフィックスとCAD/CAM/CAE雑誌ピクセル

1989年5月1日発行/毎月1日発行/通巻80号/1984年5月2日第3種郵便物認可

ISSN 0287-511X

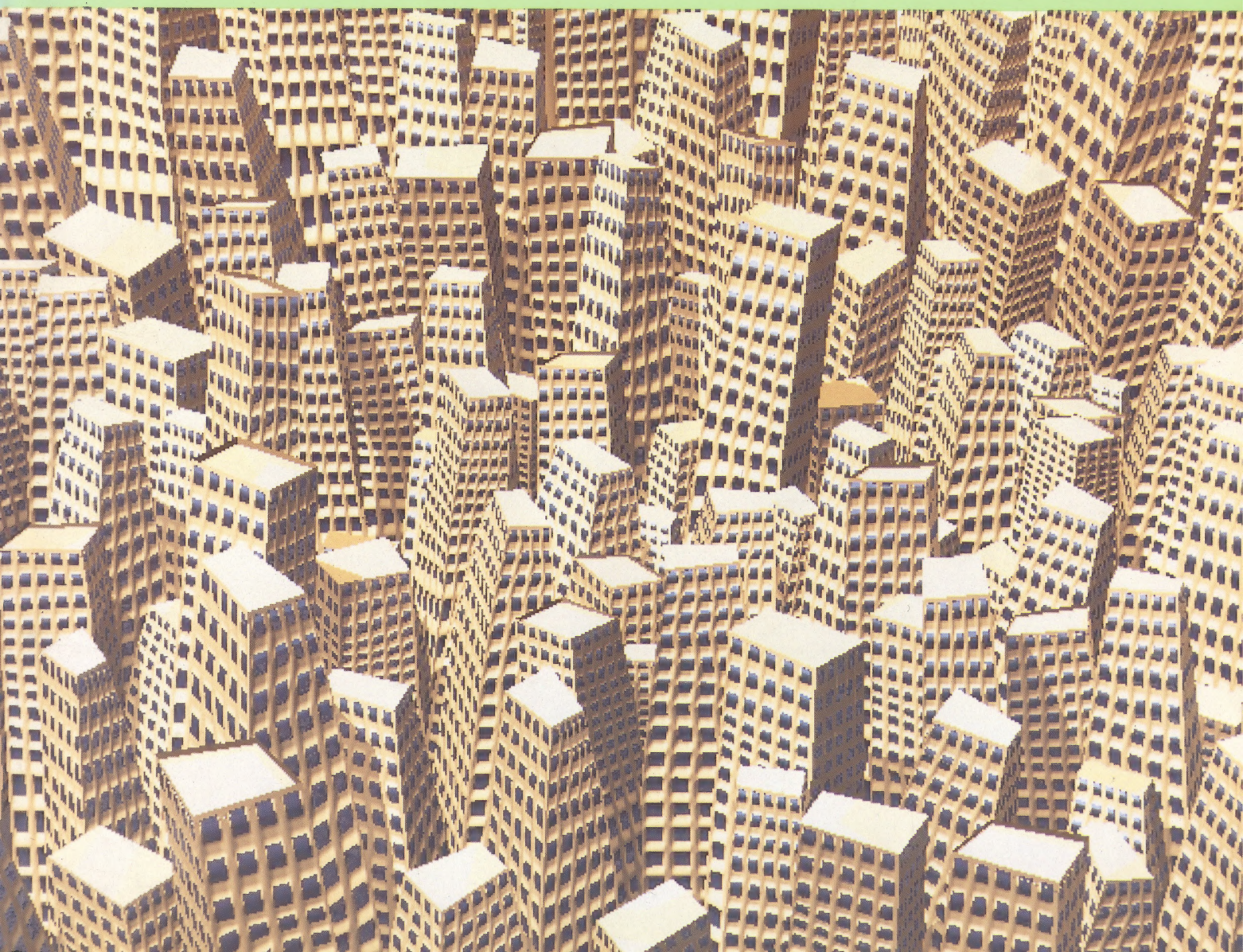
PIXEL

'89 5

No.80

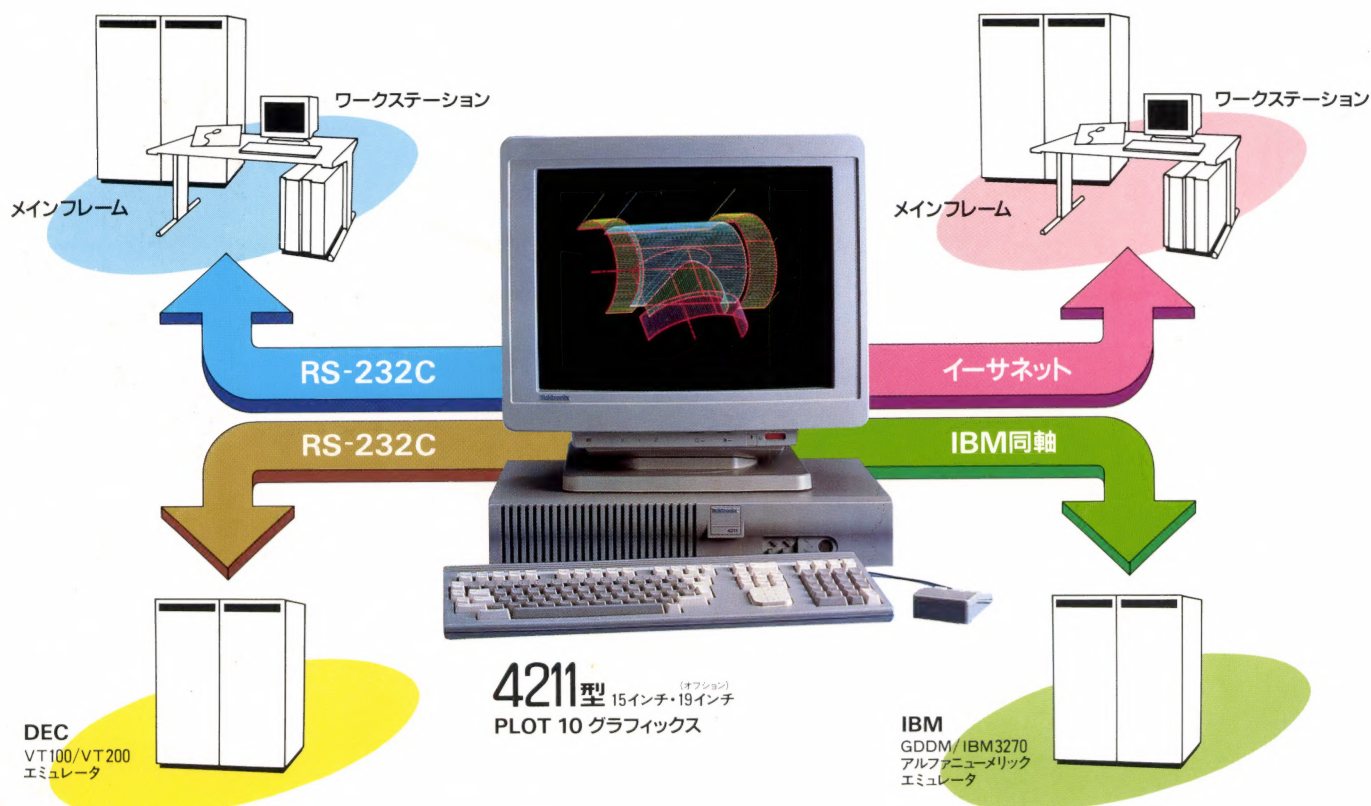
特集 グラフィック・デザインにおけるCG

特集 建築、土木のCADとCG

特集 機械設計のための計算力学

SONY Tektronix ●グラフィック世界のリーダー
ソニー・テクトロニクス
ソニー・テクトロニクス株式会社
情報機器部／東京都品川区大崎1-6-4 新大崎勧業ビル 〒141
TEL03・779・7611
仙台営業所／TEL022・267・2181 土浦営業所／TEL0298・24・2602
名古屋営業所／TEL052・581・3547 大阪営業所／TEL06・947・0321
広島地区／TEL082・247・0661 福岡営業所／TEL092・472・2626

ネットワーク カラー・グラフィック・ターミナル

●1,024×768高解像度コンパクト・デスクトップ型●40,000ベクタ/秒高速描画(当社比5倍)●セグメント・メモリ4MB(最大)●15インチ(標準)、19インチ(オプション)●低価格￥1,240,000(価格は'89年4月20日現在、消費税は別途加算されます)

ワークステーション
メインフレーム
RS-232C
イーサネット
RS-232C
IBM同軸
DEC
VT100/VT200
エミュレータ
4211型 15インチ・19インチ(オプション)
PLOT 10 グラフィックス
IBM
GDDM/IBM3270
アルファニューメリック
エミュレータ

優れたコスト・パフォーマンスを実現
4211型は新アーキテクチャ、デュアル・プロセッサ・システムの採用で高度なグラフィック処理を実現したカラー・グラフィック・ターミナルです。グラフィック・エンジン部にTI34010マイクロプロセッサと3種の独自開発のカスタムLSI、データ・マネジメント部にインテル80386SXマイクロプロセッサを採用。当社性能比5倍の高速描画、高機能等、優れたコスト・パフォーマンスを実現しました。

LANに直接接続可(オプション)
イーサネット(TCP/IPプロトコル)上のホスト・コンピュータやワークステーションに直接接続でき、ネットワーク・グラフィック・ターミナルとして使用できます。

IBM社ネットワークに接続可(オプション)
IBM社クラスタ・コントローラに同軸ケーブルで直接接続でき、IBMタイプ・キーボードでIBM3270、GDDMアプリケーションをエミュレートします。

最大32ビットのアドレス空間
約40億×40億ポイントのアドレス空間にグラフィック定義ができるため、高密度なグラフィック表示ができます。

最大256色同時表示
標準で4,096色のカラー・パレットから16色、オプションで1,670万色のカラー・パレットから256色同時表示できます。

PLOT 10コンパチブル
世界で10万本以上の実績を誇るアプリケーション・ソフトウェアとコンパチブルです。

新製品
ネットワーク
カラー・グラフィック・ターミナル
4211型

(資料請求番号 表2)

4機種そろって、さまざまな出力ニーズに対応、
バーサテックのカラープロッター。

豊富なカラー、高画質、高速出力のバーサテック・カラープロッター。いま4機種そろって、カバーする範囲は、A3サイズからB0サイズまで、ビジネスグラフから、電気、建築、LSI関連の出力までいちだんと大きく拡がりました。適材適所で、CADシステムの効率を強力にバックアップします。

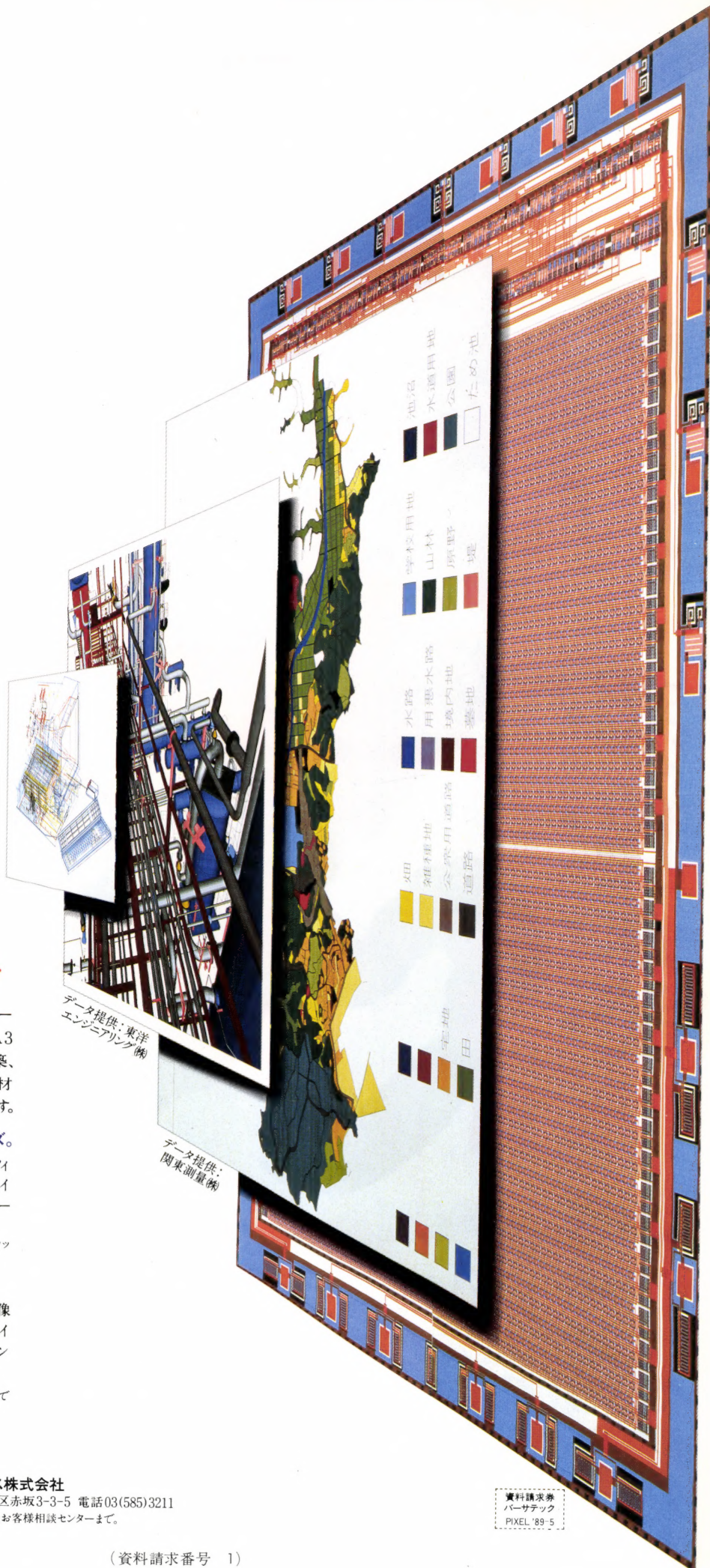
■静電カラープロッター、バーサテックCE3400シリーズ。

1インチ400ドットの高解像で、4,160色※の豊富なカラー出力。フィルムベースにもフルカラー出力。最大出力幅24インチ(A1)、36インチ(A0)、44インチ(B0)の3機種。高速・高機能コントローラーも用意。　※標準は512色。オプション・ソフトウェアにより4,160色(エリア)。

カラー出力後の紙を自在にカットするCE3436静電カラープロッター用オートカッター/スリッターも用意しています。

■サーマルカラープロッター、バーサテックC2700。

熱転写方式の卓上型カラープロッター。1インチ300ドットの高解像で、A3サイズを1枚あたり約80秒、フルカラーで出力。A3、A4サイズのカット紙使用。しかもカセット方式で、用紙交換が簡単。シンプル構造で高信頼性を実現。

★35㎜カラースライドフィルムから直接入力、最大44インチサイズまで拡大出力できる、専用のカラースライドスキャナー、ゼロックスXS-10も発売中です。

データ提供：東洋エンジニアリング㈱

データ提供：関東測量㈱

富士ゼロックス株式会社
〒107 東京都港区赤坂3-3-5　電話03(585)3211
※資料請求は本社お客様相談センターまで。

資料請求券
バーサテック
PIXEL '89-5

ビジネスショウに出展

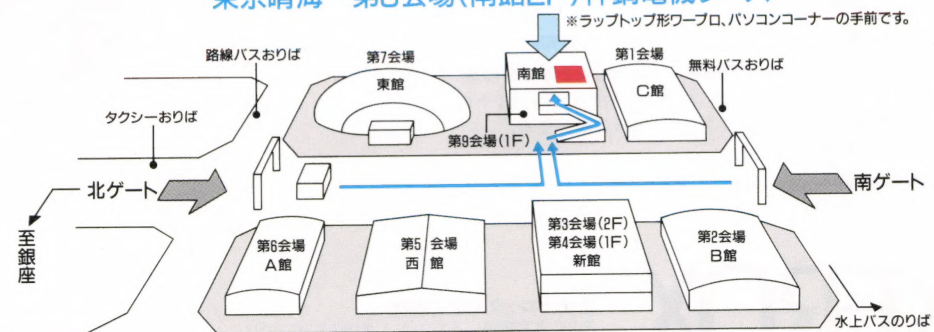
"ここを見ないとソンをする"

神鋼電機では、カラーハードコピーCHCシリーズを各種展示し、各種のグラフィックソフトによるビジュアルコミュニケーションをご紹介します。あなたが求める新しいカラーの世界が、きっとご覧いただけるはず!。
ぜひ、ご来場くださるようお待ちしています。

- 東京晴海:5月17日〜20日
- インテックス大阪:6月1日〜3日

東京晴海…第8会場(南館2F)神鋼電機ブース

※ラップトップ形ワープロ、パソコンコーナーの手前です。

路線バスおりば
第7会場
東館
南館
第1会場
C館
無料バスおりば
タクシーおりば
第9会場(1F)
北ゲート
南ゲート
至銀座
第6会場 A館
第5会場 西館
第3会場(2F)
第4会場(1F)
新館
第2会場 B館
水上バスのりば

最速兼美。
●1画面を55秒でプリントアウト。
(300dpi、SPI-3接続、バッファプリント時)
■主な特長
●熱転写方式、高解像度形
(200dpi、300dpi 高性能サーマルヘッド採用)
●27万色のスーパーマルチカラー
●1画面、55秒の高速プリント
●プリントサイズはA3、A4、B6
●セントロニクスとビデオインタフェース
(ビデオインタフェースは7.5〜130MHzに対応)
●0.5秒〜約6秒の高速データ取込み(4913色で3秒)
●ディスプレイは最大4台まで接続可能
(SPI-3-1、2、3、4使用時)
●OHPフィルムにも鮮明プリント
●NEC PC-98シリーズもワンタッチ出力
(専用アドオンボードでセントロニクス出力が可能)
CHC-335とSPI-3の組合せ
(SPI-3はビデオ信号でプリントできる高機能フレームバッファ)
神鋼 カラーハードコピー
CHC シリーズ
神鋼電機
SHINKO ELECTRIC CO.,LTD.
■お問い合せは
神鋼電機(株)端末機器本部へ
●本社/〒103 東京都中央区
日本橋3-12-2 ☎(03)274-1129
●大阪支社/☎(06)203-2244
●名古屋支社/☎(052)581-8544

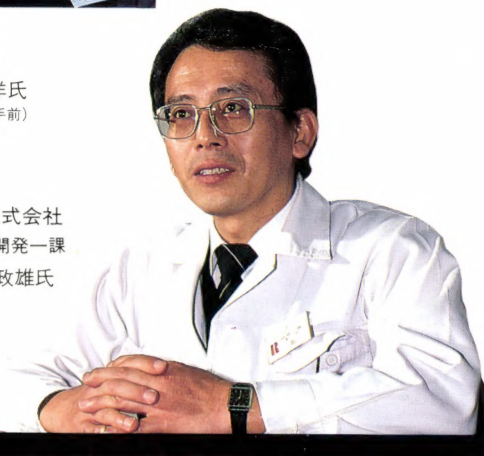
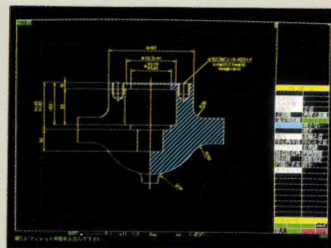
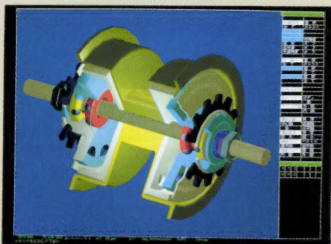
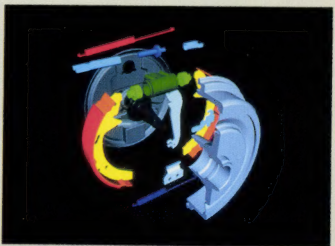
(資料請求番号 3)

CADが歴史をズームアップ。

大型製品の設計にも、使い易さで選ばれた2次元CAD「MEシリーズ10」。

もはや、複写機は私達の生活必需品として、オフィスだけではなく、暮らしの中まで用途がひろがっています。その中でちょっと特殊な複写機の世界。例えば、図書館で歴史的な書物、地図などを納めたマイクロフィルムを高精度に複写拡大するマイクロリーダープリンタや工業用大型複写機。
その設計にもYHPの2次元CAD「MEシリーズ10」が活躍しています。
今回は、OA機器の特殊分野を開拓するリコー特機株式会社殿、開発技術部の北氏と北山氏にお伺いしました。まず、ご導入のポイントは、という質問に「私どもの扱っている製品は、大型製品が多く、その設計図面は約1,000枚にもおよぶため、より早く、より確実に図面化しなければなりません」「以前は同じYHPのHP-DRAFTを使っていましたが、よりスピードアップを図るためにME-10を導入したわけです」「ME-10は大型製品を設計する場合でも、長い線を一度に引くことができ、大きなCRTでレイアウト図も描けてしまうのが便利ですね。しかし、一番の決め手は、やはり操作が簡単なことでした」というお答え。
とくに使いやすいところは、という質問には「基本コマンドだけの教育だけで誰にでも簡単に使いはじめることができる」「独自のマクロコマンドによって、特殊ネジ、長穴、止め輪などの複雑な部品の標準化が自分たちの手で推進でき、外注先との部品管理も効率よく行える」「またパーツ機能で、全体のレイアウトから部品図にスムーズに展開でき、部品表の作成も容易」などのご意見をいただきました。
現在、12システムがフル稼動し、大型製品設計の効率化に貢献するMEシリーズ10。いま、リコー特機株式会社殿で、そして皆様の暮らしのそばで役立っています。

「MEシリーズ10」
開発技術部 開発一課
係長技師　北山 敬洋氏
（写真、手前）

リコー特機株式会社
開発技術部 開発一課
課　長　北 政雄氏

〈トータルCAD/CAM/CAE を実現ーYHP CADファミリ〉

●2次元汎用CAD「MEシリーズ10」
優れた作図・編集機能に加えて、マクロ機能をはじめとする便利な機能を満載。2次元設計をスピーディに行うとともに、自動設計も可能にします。

●3次元ソリッド・モデリング・システム「MEシリーズ30」
2次元図形上での操作とわかりやすいコマンドにより、ソリッド・モデルが簡単スピーディに作成できます。

●トータルメカニカルCAE「SDRC I-DEAS」
3次元ソリッドモデラ、FEM、モーダル解析などの各モジュールを統合化。大規模な機械設計を総合的にサポートします。
※SDRC、I-DEASは米国SDRC社の登録商標です。

MEシリーズ10

横河・ヒューレット・パッカード

■資料のご請求は横河・ヒューレット・パッカード株式会社宣伝部へ――〒168 東京都杉並区高井戸東3-29-21 ●製品についてのお問い合わせは各営業所へ――本社・営業本部:03-331-6111/東部支社:03-335-8111/仙台:022-225-1011/秋田:0188-36-5021/郡山:0249-39-7111/宇都宮:0286-33-1153/関東:048-645-8031/熊谷:0485-24-6563/八王子:0426-42-1261/長野:0262-24-8012/諏訪:0266-28-8851/東京支社:03-348-4611/水戸:0292-25-7470/つくば:0298-51-5141/千葉:0472-25-7701/品川:03-458-5411/横浜支社:045-312-1252/厚木:0462-25-0031/西部支社:06-304-6021/静岡:0534-56-1771/名古屋:052-571-5171/豊田:0565-27-5611/京都:075-343-0921/神戸:078-392-4791/広島:082-241-0611/福岡:092-472-8731
AD-EMS 9015

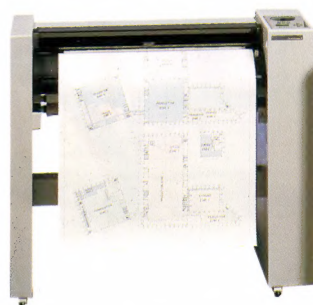
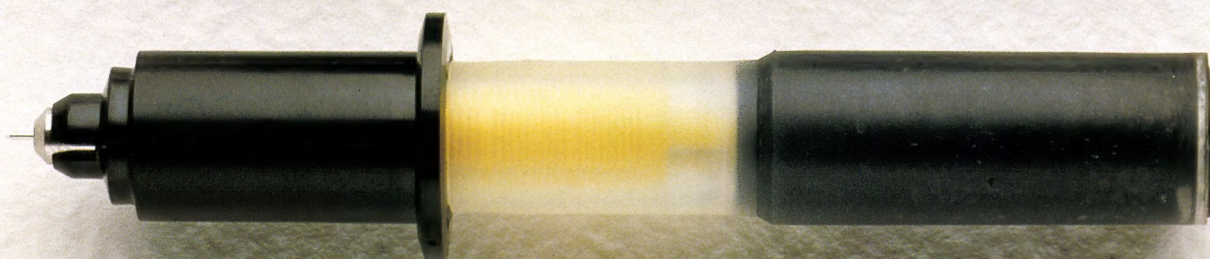
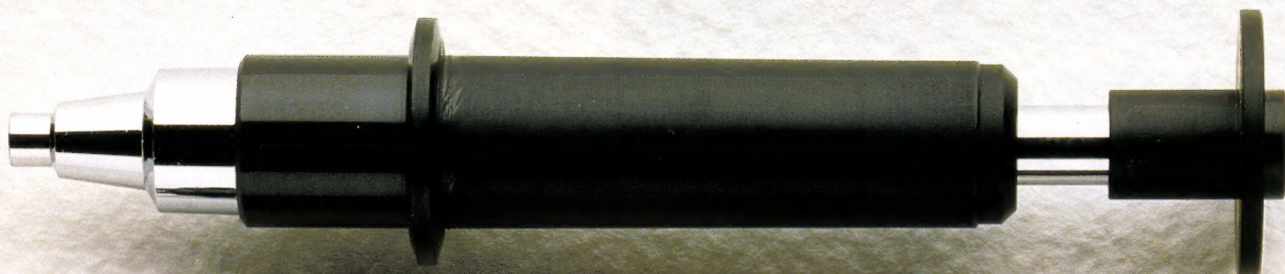
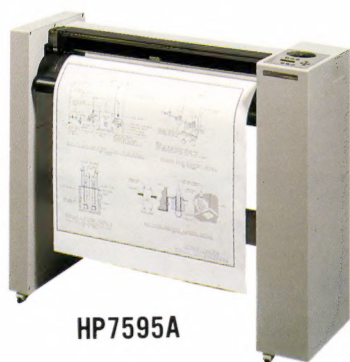
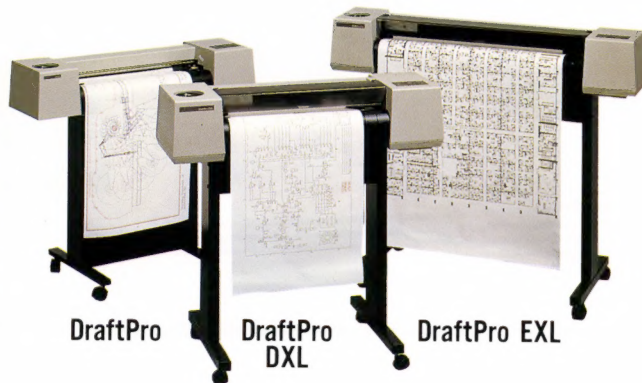
メカニカルペンシルVSイレイザブルインク

最高機種7595A/7596Aプロッタにペンシル機能を追加拡張し、ユーザー環境もますます向上しました。

トータルパフォーマンスでも、コンパクト性でも、HPプロッタファミリの中から選べます。

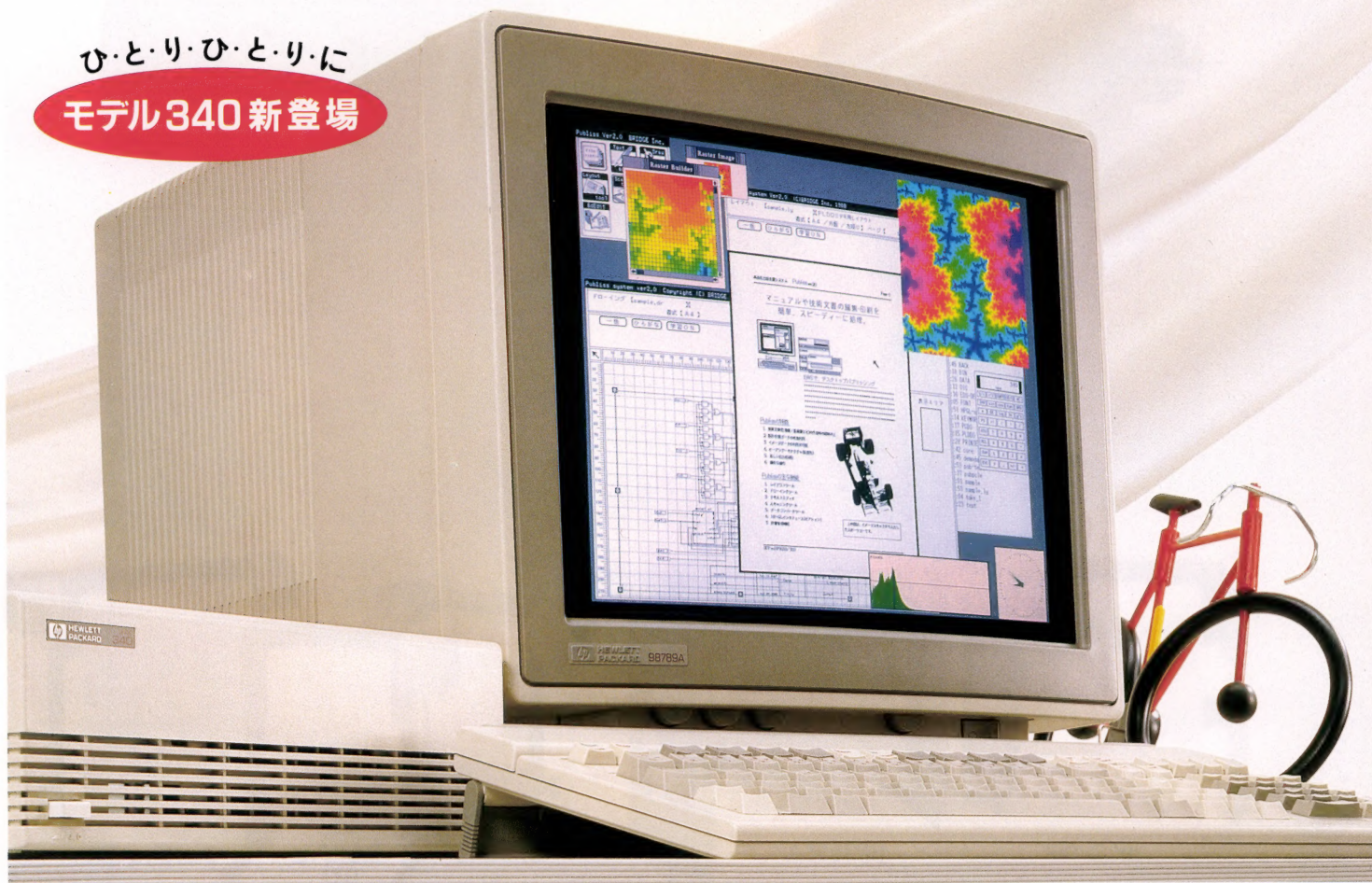
豊富なサプライ品群も合わせて自由自在に使いこなしてください。

HPプロッタファミリ

(資料請求番号　6)

HP7596A
HP7595A
DraftPro
DraftPro DXL
DraftPro EXL

YHP

ひ・と・り・ひ・と・り・に

モデル340新登場

ローエンド・マシンから
ソリッド・レンダリングマシンまで

HP9000 EWSファミリ

（資料請求番号　8）

ENGINEERING
WORKSTATION
HP 9000 EWS FAMILY

MC68030

これからのあなたに、これからのEWS、モデル340登場。

MC68030搭載―HP9000シリーズ300 EWS新ラインアップ完成!!

モデル340誕生。エントリ・レベルながら、最新のマイクロプロセッサMC68030を搭載し、4MIPSのコンピューティング・パワーを実現しての登場です。従来のエントリ・マシンにくらべ約2倍のプライス・パフォーマンス(当社比)を実現。しかも、2次元モノクロから3次元ソリッドまで6タイプ(M、MH、C+、CH、CHX、SRX)のグラフィクスが選択できる充実したラインアップ。パブリッシング、ソフト開発、ドラフティング、回路入力、プリント基板/LSI設計、さらにソリッドモデリングなど多彩なアプリケーション・ニーズに自在にお応えします。また、ネットワーク・ソリューションへの対応も万全、ThickLAN、ThinLAN、Star LAN10から選択可能。さらに、スロット増設のオプションも用意し、多様なアプリケーションも可能にするキャパシティを誇ります。しかも、共通OSにUNIX[*1](HP-UX)を採用したほか、ネットワーク(IEEE802.3/Ethernet[*2]、TCP/IP、ARPA/Berkeleyサービス、NFS[*3]等)、ウィンドウ・システム(X.11[*4])、グラフィクス(PHIGS、GKS、CGI、CORE等)にも業界標準を徹底採用しています。

● ● ●

モデル340の登場により、先にリリースのハイ・パフォーマンス機モデル360/370とあわせてMC68030搭載のHP9000シリーズ300EWSは、3モデル・18タイプの強力なラインアップを完成させました。

■HP 9000 EWSファミリ性能一覧表

(性能:シリーズ800 モデル835、モデル825/シリーズ300 モデル370、モデル360、モデル340/価格)

モデル名	CPU(MHz)	メモリ(MB)	グラフィクス・オプション M 1024×768×1	MH 1280×1024×1	C+ 1024×768×6	CH 1280×1024×8	CHX 1280×1024×8	SRX 1280×1024×8~24	TurboSRX 1280×1024×8~24	OS
835	HPPA(RISC)	8~96					●	●	●	HP-UX (UNIX System V.3+4.2/4.3BSD)
825	HPPA(RISC)	8~96					●	●		
370	MC68030(33)	8~48		●	●	●	●	●	●	
360	MC68030(25)	4~16		●	●	●	●	●	●	
340	MC68030(16.6)	4~16	●	●	●	●	●	●		

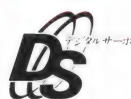
*1. UNIXは、米国AT&Tベル研究所で開発されたオペレーティング・システムです。
*2. Ethernetは、XEROX社の登録商標です。
*3. NFSは、Sun Microsystems社の登録商標です。
*4. X.11は、米国MIT(マサチューセッツ工科大学)の登録商標です。

■資料のご請求は横河・ヒューレット・パッカード株式会社宣伝部へ――〒168 東京都杉並区高井戸東3-29-21●製品についてのお問い合わせは各営業所へ――本社・営業本部:03-331-6111/東部支社:03-335-8111/仙台:022-225-1011/秋田:0188-36-5021/郡山:0249-39-7111/宇都宮:0286-33-1153/大宮:0486-45-8031/熊谷:0485-24-6563/八王子:0426-42-1261/長野:0262-24-8012/諏訪:0266-28-8851/東京支社:03-348-4611/水戸:0292-25-7470/つくば:0298-51-5141/千葉:0472-25-7701/品川:03-458-5411/横浜支社:045-312-1252/厚木:0462-25-0031/西部支社:06-304-6021/静岡:0534-56-1771/名古屋:052-571-5171/豊田:0565-27-5611/京都:075-343-0921/神戸:078-392-4791/広島:082-241-0611/福岡:092-472-8731

AD-TCG9055

(資料請求番号 9)

40th ANNIVERSARY
おかげさまで40周年

衝撃のデビュー！

DS デジタルサーボ

デジタルサーボ駆動方式。この高級仕様を
小型機・実用機に搭載して、いま衝撃のデビュー！

DS（デジタルサーボ）& FDD（1Mバイト2HD）搭載！
全身ダイナミックに進化して新登場!!

A3

マイプロット4000シリーズ

プロのスペックをパーソナルユースに。

●高速(640mm/s)・高描画品質(0.005mm)を実現するデジタルサーボ駆動方式●3.5インチのFDD(1Mバイト2HD)搭載●LCDによる対話式操作●漢字作図(新JIS第1・第2水準)●16ビットCPUによる高級インテリジェント(ペンソーティング)機能

品　番	価　格	駆動方式	FDD	用紙固定	LCD
MP4400	¥268,000	DS	○	静電	○
MP4300	¥198,000	DS	—	静電	○
MP4200	¥148,000	パルス	—	静電	—
MP4100	¥128,000	パルス	—	マグネット	—

※以上スペックは、機種により異ります。〈カタログをご覧下さい。〉

'89.3.1～5.31

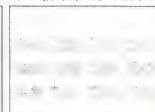
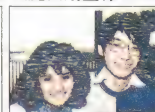
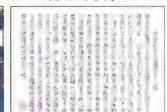
期間中お買い上げいただき、アンケートを返送の方にダブル・プレゼントを進呈!!
①もれなく、オリジナルテレホンカード(50度数)を。
②さらに、抽選で30名の方にシステム電子手帳を。

第2回 マイプロット作品コンクール入選作品決まる！

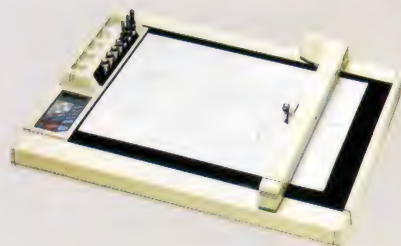
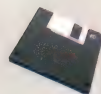
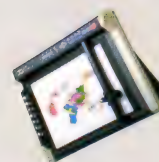
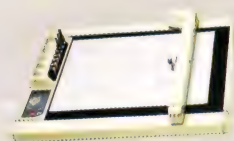
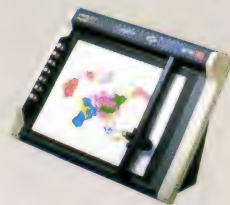
●佳作　小山純一様(神奈川)他9名様。
多数のご応募ありがとうございました。

■最優秀賞
関野ひかる様(東京)
☆「ミスターの年賀状」

■優秀賞
(福島) 沖田敏行様・村岡綾子様
☆「テレホンカードのための十二星座」

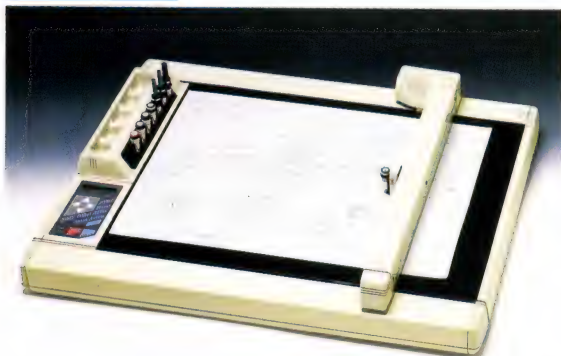
■優秀賞
入鹿山剛堂様(茨城)
☆「FRIEND」

■努力賞
西久保和男様(東京)
☆「般若心経」

—情報を描く—

GRAPHTEC

デジタルサーボ

DS搭載。

先進のハイコストパフォーマンスを実現して新登場!!

A1・A2

A1・A2プロッタ7000シリーズ

ハイテクノロジーをパーソナルユースに。

●鉛筆使用可能●高速(600mm/s)・高描画品質(0.005mm)を実現するデジタルサーボ駆動方式●LCDによる対話式操作●漢字作図(新JIS第1・第2水準)●16ビットCPUによる充実のインテリジェント(ペンソーティング・ベクトルソーティング)機能

品　　番	価　　格
FP 7100(A1)	¥790,000
FP 7200(A2)	¥640,000

グラフテック株式会社

本社/〒108 東京都港区三田3-13-16 三田43森ビル16階

●カタログのご請求は、本社広報宣伝部まで。

お問い合せは●プロッタ事業部 TEL.0466-87-4111(代)

見て、触れて、体験できるグラフテック総合ショウルーム

■PLAZA TOKYO/OSAKA■

電話によるご相談は 03-453-0512 インフォメーションセンター

プラザ東京 月~金 AM9:30~PM5:30 東京都港区三田3-13-16三田43森ビル16F(本社内) TEL.03-453-0512 JR田町駅、都営地下鉄三田駅下車 徒歩5分

プラザ大阪 月~金 AM9:30~PM5:30 大阪市淀川区西中島5-14-5新大阪INビル8F(関西営業所内) TEL.06-305-1381 JR新大阪駅、地下鉄御堂筋線新大阪駅下車 徒歩2分

●北海道営業所 011-232-1332 ●東北営業所 022-236-3405 ●関東営業所 0485-25-2381 ●茨城営業所 0298-58-1211 ●東京営業所 03-444-7011 ●公需営業所 03-444-8281 ●秋葉原営業所 03-834-5688

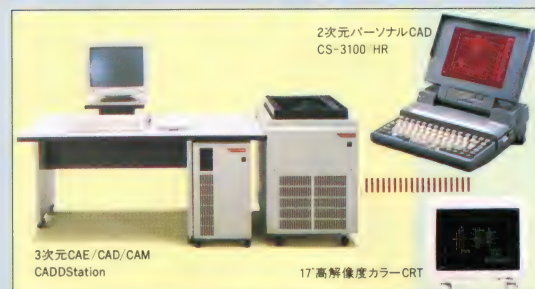
●多摩営業所 0423-25-5050 ●神奈川営業所 045-541-6811 ●厚木営業所 0462-22-4971 ●中部営業所 052-776-0821 ●関西営業所 06-305-1381 ●西部営業所 082-261-2931 ●九州営業所 092-474-2441

(資料請求番号 11)

資料請求券 ピクセル 5月号

KUBOTA COMPUTER INC.
TITAN発、リアルタイム・グラフィックス現象。
〝思考と直結したグラフィックス・コンピューティングが、創造性をいかんなく刺激し、生産性を飛躍的に高める…〟
リアルタイム・グラフィックス現象を体験したエンジニアは、TITANの出現によってエンジニアリング新世紀が到来したことを確信した―― クボタコンピュータのグラフィックス・スーパーコンピュータ『TITAN(タイタン)』は、スーパーコンピュータに匹敵する演算パワーと高速3次元グラフィックス機能を結合。従来のいかなるスーパーコンピュータも成し得なかった、科学技術計算のリアルタイムな視覚化を実現しました。しかも、エンジニア一人一人の占有を可能にする低価格で!その適用範囲は幅広く、CAE・分子設計・流体解析・CGアニメーション・画像処理…等の、様々な分野でインタラクティブ・コンピューティングの効果を目のあたりにみせてくれます。
Graphics Supercomputer
TITAN
●ピーク演算性能:16~64MIPS、16~64MFLOPS(1~4プロセッサ) ●グラフィック性能:5千万グローシェーディドピクセル/秒、40万ショートベクトル/秒、7.5~20万ポリゴン/秒
クボタコンピュータ株式会社
〒160 東京都新宿区新宿2-8-8 ☎03(225)0741
大阪支店 〒541 大阪市中央区本町4-4-25 ☎06(264)2501
名古屋営業所 〒450 名古屋市中村区名駅3-22-8 ☎052(564)5091
山梨工場 〒400-02 山梨県中巨摩郡白根町下今諏訪907-8 ☎0552(84)4861

(資料請求番号 13)

TOSHIBA
個性を伸ばすと、会社も伸びる。
技術者の個性と創造力を伸ばすには、時間のゆとりも欠かせません。
導入しやすく、拡張性に富んだCS-CADDSネットワークは、
技術者1人に1台のゆとりを実現。仕事の質と一緒に、会社も伸ばします。
東芝の優れたEWS技術と世界中で豊富な実績を持つCADDSソフトウエアが融合しました。3次元モデリング・解析など高度な作業を支援するCADDStationと手軽に2次元設計・製図を行うパーソナルCAD CS-3100/HR。この2つのシステムを結んだCS-CADDSネットワーク。技術者の個性と創造性を確実にシステムに活かせます。
■CAD間のデータを共有
CADDStationとCS-3100/HRで、双方向のデータ交換が可能です。このCS-CADDSネットワークには、転送速度10Mbpsという高速のイーサネット, TCP/IPを採用。大量データの転送もスピーディにできます。
■資源の共有化
プリンタ、プロッタ、及びハードディスク等の資源を共有。ネットワーク内で効率のよいシステム構築ができます。
■機能分散
業務の形態に応じ2次元作業、3次元作業の分担ができます。個々のシステムをフル活用する組み合わせをフレキシブルに構成できます。
■CPU分散
コンピュータ資産を活かしながら、必要に応じてシステム拡張ができるAdd-on方式です。
※CADDS:Computervision Automated Design and Drafting System
※CADDSはCOMPUTERVISION社の開発ソフトです。
2次元パーソナルCAD
CS-3100 HR
3次元CAE/CAD/CAM
CADDStation
17"高解像度カラーCRT
東芝CAE/CAD/CAM
CADDStation/CSシリーズ
お問合せは 株式会社 東芝/コンピュータ事業部 電話(03)457-2725(ダイヤルイン)
産業システム事業部 電話(03)457-4340(ダイヤルイン)
〒105 東京都港区芝浦1-1-1(東芝ビルディング)
本誌面に掲載しております全商品及び役務等をご購入の際消費税が付加されますのでご承知おき願います。
先端技術をオフィスに…E&Eの東芝
エネルギーとエレクトロニクス
(資料請求番号 14)

CADIX's POWER

超最先端テクノロジーの凝縮

CADIX

大量既存図面からCADデータまで、あらゆるデータの光ディスク一元管理を実現し、入力、修正、編集から通信までをカバーするCADIX最上位システム。

統合情報ネットワークシステム FX-7001

1. 大型図面からCADデータまで技術情報を一元管理
大型設計図面、CADデータベース、その他あらゆる技術情報を大容量の光ディスクライブラリーで統合一元管理。あらゆるデータの保存・管理にまつわる様々な問題点を一挙に解決！

2. 新次元ラスターCAD機能
大型（A 0）既存設計図面の設計変更／編集、新規設計にラスターCAD／ベクターCADが対応。ラスター／ベクター統合データベースをもつスーパーワークステーションS-21によりCADデータ変換もOK！

3. パーフェクトな図面検索機能
各ユーザ既存の管理方法に対応した図面検索体系への置き換えも可能。綿密な打ち合せによりお客様の満足のいく検索システムを作成！

4. 幅広いコミュニケーション機能
他システムとの容易なコミュニケーションを可能とするイーサネット（TCP／IP）をサポート。また、高速デジタル回線による広域、長距離間での高速スループットを実現！

5. 新世代ハードウェアを搭載
90ギガバイト光ディスクライブラリー、スーパーワークステーションS-21を中心に読み取りスピード30秒／A1の高速スキャナー、A 0判ラスター／ベクターを同時出力可能な高速静電プロッタなど様々な新世代ハードウェアを搭載！

お問合せ先　東京：03(427)0401　名古屋：052(212)1551　大阪：06(947)5521

株式会社 キャディックス

本　社	〒154　東京都世田谷区新町2-26-15	TEL.(03)427-8901	FAX.(03)420-6084
営業本部・東京ショールーム	〒154　東京都世田谷区桜新町1-12-10 6F	TEL.(03)427-0401	FAX.(03)427-0201
名古屋支社・ショールーム	〒460　名古屋市中区栄2-3-1 広小路ビル12F	TEL.(052)212-1551	FAX.(052)212-1531
大阪支社・ショールーム	〒540　大阪市中央区城見2-1-61 Twin21 MIDタワー8F	TEL.(06)947-5521	FAX.(06)947-5535
R & D	〒154　東京都世田谷区桜新町1-12-10 5F	TEL.(03)439-0821	FAX.(03)439-0640

（資料請求番号　17）

NEC PC-9801シリーズ、IBM PC/AT、
富士通 FMRシリーズ対応。

まだ、こんな道具だけに頼っていませんか？

"YUI"が、ペーパー・セメントやカッターにとってかわることはないかもしれません。しかし、そのカット＆ペースト機能を使えば任意形状領域の拡大、縮小、回転、変形などが可能になります。ズーム、ウィンドウ・メニューなどグラフィック・ディスプレイ・コントローラ"FBX24"の豊富なハード・ウェア機能と1024×1024×24Bitのメモリー領域をフルに活用した多彩なペイント機能をアート・ワーク、デザイン・ワークなどにご活用ください。また、当社ショー・ルームにて常時デモを行なっておりますので、ぜひ一度ご自身の目で"YUI"の操作性をお確かめください。

■標準価格 ￥200,000.-

- ハードウェアによるリアルタイム・ズーム/スクロール
- 640×480ピクセルのペイント領域
- 任意形状領域のカット&ペースト
- 画像の自由な拡大、縮小、回転、変形、移動
- テキストの拡大、回転、移動機能
- 形状記憶式図形描画機能
- 色相変換機能
- ボカシ機能
- 消しゴム、UNDO機能
- マスキング機能
- 標準入出力サポート
  カラー・スキャナー：SHARP JX-200、JX-450
  カラー・プリンタ：CANON FP-510

フルカラー・ディスプレイ・コントローラ"FBX24"

開発・発売元/株式会社テクネ 〒150 東京都渋谷区宇田川町2-1、No. 620 ☎03-464-6927 FAX. 03-476-2372

情報は友だち
RICOH

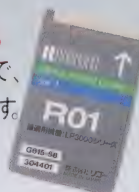
HP-GL対応・A3判出力のレーザーだから、

設計図面の中間チェックや
打ち合わせ時に威力を発揮。
各社のパソコンに接続できる、
高性能レーザープリンター
"PC LASER"ついに誕生。

LP3320-SP4

標準システム価格……………628,000円
(LP3320-SP4、エミュレーションカード、フォントカートリッジF24Mを含む)

●HP-GLに対応。プロッターよりも高速度。
プロッター制御用の命令セットHP-GL(Hewlett Packard Graphic Language)に対応。高性能レーザープリンターならではの高速度で、細かい線までくっきりと出力することができます。

●エミュレーションカードでさまざまなパソコンに対応。
エミュレーションカード(ICカード)を差しかえるだけで、各社のパソコン用プリンターとして、威力を発揮します。
適応機種:NEC PC-9800/N5200、IBM PS55、TOSHIBA J3100、日立 B16、リコーMr.マイツール

●お手持ちのソフトの変更は不要。
現在お使いのドットプリンターとプラグコンパチブル。適応機種に対応するほとんどの市販ソフトは、そのまま使えます。

●A3判原稿のB4・A4縮小印字も自由自在。
プリンター上で設定できる縮小サイズは5種類。操作は、パネルの縮小キーを押すだけなのでとても簡単。ご希望のファイリングサイズで、鮮明なプリントアウトが行えます。

●フォントカートリッジひとつで対応できる豊富な文字フォントも魅力。

パソコン用レーザープリンター

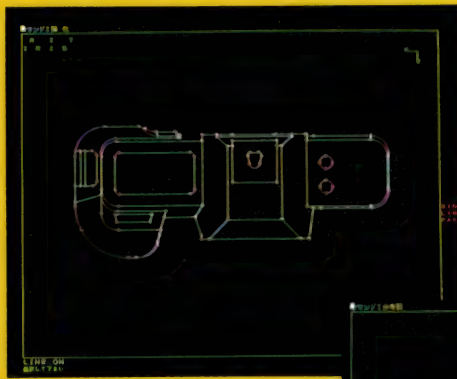
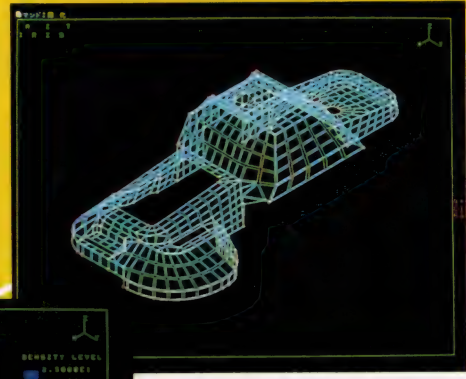
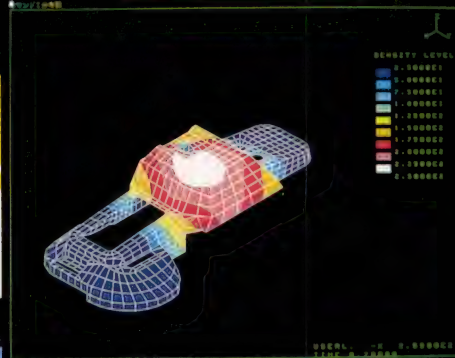
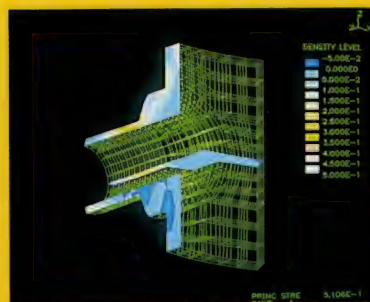
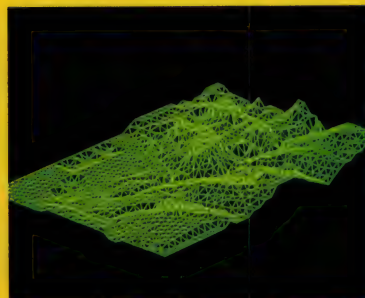
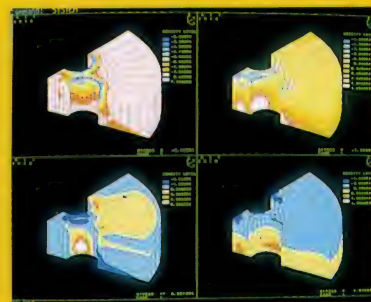
PC LASER

●カタログを差し上げます。ハガキに資料請求券を貼って右記住所までお気軽にお送りください。〒107 東京都港区南青山1-15-5 株式会社リコー宣伝部 リコー
また、電話によるお問い合わせは☎03(578)3111 I/O営業部計画課までお気軽にどうぞ。
(資料請求番号 19)

資料請求券
LP3320-SP4
PIXEL
4/20号

操作性を重視した対話型統合CAEシステム

IRIS

IRISはエンジニアリング業務の効率化を図るため、設計から解析・評価さらには製図、技術レポート作成と幅広くサポートするソフトウエアパッケージ群です。IRISは、対話形式でポップアップ・メニューをマウスあるいはペンで指示することで必要な機能を操作でき、また形状の生成においても、ディスプレイ上に表示されている形状あるいはコマンドを指示することでシステムを操作できます。

株式会社 **エイ・アイ・テクノロジー**
〒101 東京都千代田区東神田2-6-2 タカラビル9F
TEL 03-864-6912㈹

男女ソフトウエアエンジニア募集中
コンピュータグラフィックスに興味のある方は履歴書を送付、またはご連絡下さい。 担当 広瀬

■販売代理店
センチュリリサーチセンタ㈱ 営業開発第1部 〒103東京都中央区日本橋本町3-6-2 TEL 03-665-9741
ニチメンデータシステム㈱ 営業部 〒103東京都中央区日本橋本町1-3-2共同ビル TEL 03-241-2611
メモレックス・テレックス㈱ EWS営業部 〒103東京都中央区八重洲1-7-20 八重洲口会館 TEL 03-275-0452
東芝エンジニアリング㈱ EWSシステム部 〒210 神奈川県川崎市幸区堀川町66-2 興和川崎西口ビル TEL 044-548-3354
日本鋼管㈱ コンピューター・機器営業部 〒102 東京都千代田区九段北4-1-3 飛栄九段北ビル TEL 03-222-5965
キヤノン販売㈱ 特機営業部 〒108東京都港区港南2-13-29キヤノン・オプティカルセンター TEL 03-740-3334
日本電子計算㈱ 科学技術事業部 〒103東京都中央区日本橋兜町6-7 TEL 03-668-6171

真 mana 名

NCGA’89
Philadelphia Civic Center (Hall A)
April 16—20

美しく、自在に。
アウトライン・フォントが進化した。

ハッチング処理をした
アウトライン・フォント

作成された
アウトライン・フォント

文字元パターン

優 ← 優 ← 優

高品位フォントをパソコン上で作成する「真名」、日本語DTP時代のニーズに応えて、新登場。

パソコン上で稼働

真名はパソコン上でアウトライン・フォント作成を実現します。特別なハードウェアは必要なく、従来のようにミニコンやEWSを使用する場合に比べ、低コストです。

ドット・パターン編集機能

スキャナー、TVカメラよりドット・パターンを入力できます。マウスにより読み込んだパターンに追加、修正が可能で、スムーズ拡大機能や入力ノイズの修正機能など高度な編集機能も装備しています。

円弧、ベツェール曲線のあてはめ

真名では、特殊なアルゴリズムの開発により、曲線部の円弧、ベツェール曲線のあてはめを高速に自動的に行います。ドット・パターンでは不可能な美しいアウトライン抽出を実現しました。

アウトライン編集機能

作成されたアウトラインは、直線、円弧、ベツェール曲線として修正、追加が可能です。文字の大きさの自由な選定、回転、白抜きなども容易に実現。パソコンCADクラスの編集機能を持っています。

バッチ処理機能

ユーザーが設定したバッチ・プログラムにより、同一処理の多いアウトラインの抽出を自動実行することが可能です。

カッティングプロッターに対応

作成されたアウトラインは、カッティングプロッターで出力できます。ポップ文字やロゴタイプの作成はもちろん、各種デザイン作業に威力を発揮します。 ※下記価格に消費税は含まれていません。

価格 150万円(MS-DOSは別売)

使用条件

●パソコン:NEC PC-98XA、PC-98XL、PC-98XL²
入力装置:NEC PC-IN502およびTVカメラ等
CRT:ハイリゾリューションディスプレイ
●メモリ 768Kバイト実装 ●MS-DOS 3.0以上
●マウスが必要です。ただし、アウトラインの抽出部分のプログラムには必要ありません。 ●アウトライン・フォントの編集プログラムについては80287等のNDPが必要です。
※MS-DOSは米国マイクロソフト社の登録商標です。

EP’89に出展

■会期:4月20日(木)〜22日(土)
■会場:池袋サンシャインシティ ミプロ展示場

※お問い合わせ
(株)インフォメーション アンド コントロール研究所 03(352)4746

総発売元
Mitsubishi Corporation
三菱商事株式会社
取扱セクション
技術部インフォメーション テクノロジーチーム
〒100-86 東京都千代田区丸の内2丁目6番3号 TEL.03(210)2538、7543、7386

開発元
ICL
Information and Control Laboratory Co., Ltd.
株式会社インフォメーション アンド コントロール研究所
〒160 東京都新宿区新宿5丁目11番22号中島ビル5F TEL.03(352)4746 FAX.03(357)7114

NEW

インテリジェントカラーコピー

5600/5900シリーズ

ビデオ4096色、ホスト制御なら色に制限のないカラー熱転写全7モデル。

●サイズはA4専用、A4/A3共用●解像度200DPI(0.127mm) ●75秒/A4の高速出力●RGBビデオ信号は一秒以下で受信●ペン/カラー静電プロッタ用データ、およびVDIフォーマットへの対応(5900)●全機種、エプソンFP80エミュレーション

即戦力を発揮する静電プロッタ

5700/5800シリーズ

NEW

モノクロなら350万円、カラーでも930万円で即稼働。

倍速モードを備えた超高速モノクロ5700シリーズ。独自の色ずれ防止機構により正確な色彩表現力を発揮するカラー5800シリーズ。●400DPIの高解像度 ●コントローラ内蔵 ●24″/36″/44″幅の各3タイプ ●標準25MB(5700)、12MB(5800)の大容量ディスク内蔵 ●チャンネル、イーサネットなど豊富なインターフェース●オートクロスカッタ(オプション)

速度・信頼性を極めたペンプロッタの一例

M-1023

NEW

コストパフォーマンスの究極。驚異のA1対応プロッタ。

機械的最高速度107cm/秒、最大加速度2.8Gに加え、内蔵のPMファームウェアはペンチェンジとペンムーブの無駄を省いてトータルスループットを飛躍的に向上。
●用紙はA4～A1任意(自動サイズ識別)
●解像度0.0127mm ●8本ペンセレクション
●入力コマンドはCalComp960、907(PCI)、およびHPGLエミュレーション
●標準価格890,000円

PRICE EXAMPLE

プロッタ主要製品価格

モデル	本体標準価格	内容	
M-1023	890,000円	A4～A1任意大カット紙対応グリットホイールプロッタ	
M-1045SP	1,200,000円	A4～A1任意大カット紙対応、鉛筆標準仕様ベルトベッドプロッタ	
M-1044GT	2,180,000円	最大36mのロングプロットをはじめA4～A0のカット紙にも対応	
M-5602	680,000円	CGアプリケーションに最適な低価格・汎用A4カラー熱転写プロッタ	
M-5912	1,775,000円	カラー熱転写プロッタともフルコンパティブルなA4/A3カラー熱転写プロッタ	
M-5725	3,500,000円	24″(A1)モノクローム静電プロッタ	25MBディスクメモリとインターフェース込みのシステム価格
M-5735	4,500,000円	36″(A0)モノクローム静電プロッタ	
M-5825	9,300,000円	24″(A1)カラー静電プロッタ	12MBディスクメモリとインターフェース込みのシステム価格
M-5835	11,300,000円	36″(A0)カラー静電プロッタ	

鉛筆標準仕様
低価格・高性能・ベルトベッドプロッタ

M-104XSPシリーズ

鉛筆プロット対応で、用途を大幅に拡大。

●標準価格:A1/1,200,000円
A0/1,450,000円

OEM 特約店募集中

日本カルコンプ株式会社
〒104 東京都中央区入船3-7-1 ☎(03)555-8911
大阪(06)304-2012・名古屋(052)951-8531・福岡(092)474-5761

製品に関する資料請求は
0120 フリーダイヤル 0120-41-8911

SOURCE
MULTIPLE COPY
Boundary Representation of Solids
MATROX News

パソコンがEWSのグラフィックに肉迫する。

高速・鮮明・高解像度タイプ
Personal HOOPS TYPEⅢ登場

神戸製鋼のパソコン用3次元リアルタイムグラフィックエンジンPersonal HOOPSは、もっとビジュアルに、もっとパーソナルにと求められる3次元グラフィックスをパソコンの世界で実現してきました。
さらに神戸製鋼は、ハイレゾルーションタイプを開発し、現代の要求に応えます。
Persoral HOOPSは、インテリア、アパレルなどの各種デザインや、CGフィルムなどで求められる高精度・高品質なグラフィックスを提供します。

NEC PC98シリーズの拡張バス・スロットに Personal HOOPSを実装するだけで、PC98シリーズ上で、3次元グラフィックスがリアルタイムに実現できます。

TYPE Ⅲ による
デモンストレーション画面

仕様

MODEL		TYPE II	TYPE III
DISPLAY RESOLUTION		640×400 ※1	1120×750 ※1
FRAME BUFFER		640×400×12bit 2 plane	1120×750×12bit 2 plane
GRAPHIC COLORS		4096 ※2	4096 ※2
GRAPHIC PROCESSORS	3D COORDINATES PROCESSOR	T800 20MHz 10MIPS 1.5MFLOPS	T800 20MHz 10MIPS 1.5MFLOPS
	DRAWING PROCESSOR	T414 20MHz 10MIPS 125000vector/sec	T414 20MHz 10MIPS 125000vector/sec
ON-BOARD MEMORY	LOCAL MEMORY / DUAL PORT MEMORY	2MB ※3 / 4KB	2MB ※3 / 4KB
BUS INTERFACES		NEC PC9800 EXPANTION SLOT	
SOFTWARE		HOOPS™, HOOPS server (for LATTICE-C, MS-C)	

TYPEⅡ、Ⅲは外部電源が必要
アクセサリー：D-RGB入力CABLE、A-RGB出力CABLE
※1 アナログRGBディスプレイが必要(パソコン出力とスーパーインポーズ可能)
※2 オプションで1670万色再現可能
※3 オプションで8MBまで増設可能
HOOPSは、Ithaca Software社の登録商標です。
HOOPSの、日本での総販売代理店はファモティック株式会社です。

パソコン用3次元リアルタイム高速グラフィックエンジン

新事業本部　情報・エレクトロニクス部

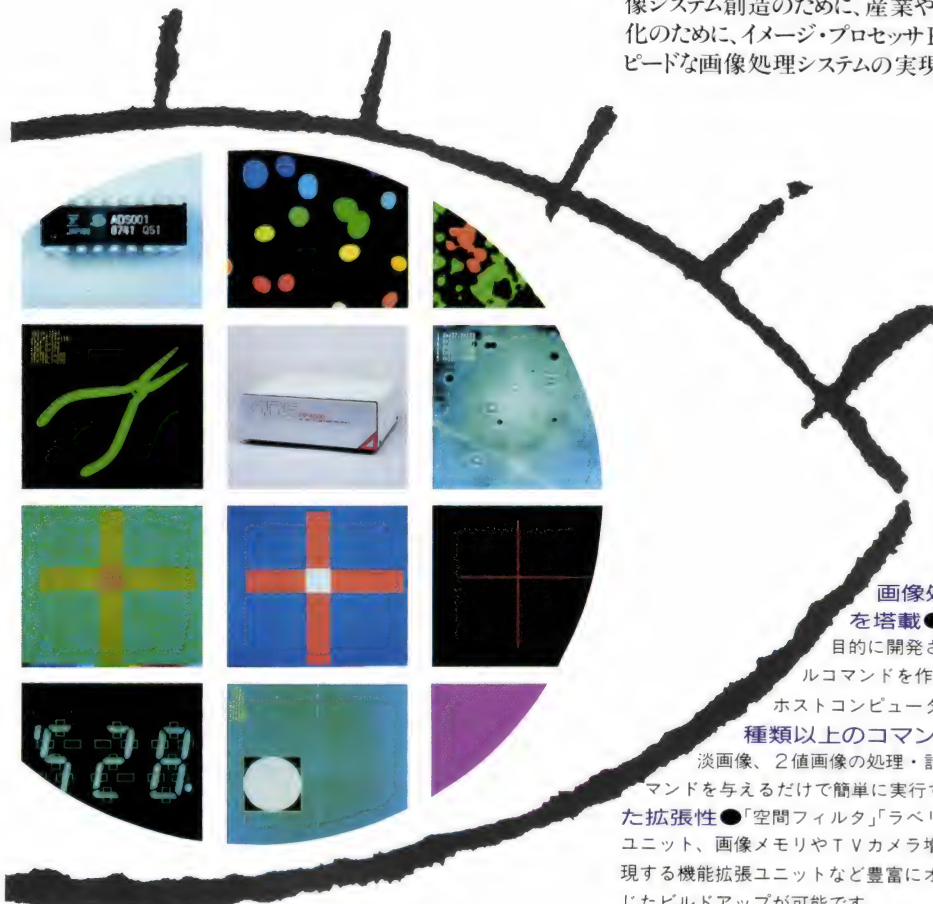
東京　〒108東京都港区芝浦4丁目9番25号芝浦スクエアビル8F　TEL.(03)5476-9221　FAX.(03)5476-9280
神戸　〒651神戸市中央区脇浜町1-3-18テクノピア　TEL.(078)261-5291　FAX.(078)261-5296

(資料請求番号　24)

ADS

イメージ、開眼。

「見たい」を「見える」に変えるPIP-4000

ADSの画像処理技術はまさにイメージ創造と呼ぶにふさわしいテクノロジー。人の眼を越える機能の実現です。その未踏の領域に近づくために、画像処理用言語であるPIPLを搭載し、一段とマンマシン・インターフェイスを強化。よりやさしいオペレーションを実現しました。画像システム創造のために、産業や医用分野などの計測・検査の自動化のために、イメージ・プロセッサPIP-4000がフレキシブルでハイスピードな画像処理システムの実現に貢献します。

画像処理専用言語PIPL(PIP-Language)**を搭載**●あらゆる画像処理の効率化・簡素化を目的に開発されたPIPL(ぴっぷる)を搭載。オリジナルコマンドを作成して、PIP-4000にダウンロードでき、ホストコンピュータ側の負荷軽減をはかります。**150種類以上のコマンドを標準装備**●ＴＶ画像の入力、濃淡画像、２値画像の処理・計測などが、ホストコンピュータからコマンドを与えるだけで簡単に実行することができます。**目的に合わせた拡張性**●「空間フィルタ」「ラベリング」等の高速化を図る高速画像処理ユニット、画像メモリやＴＶカメラ増設、スローアナログ信号の入力等を実現する機能拡張ユニットなど豊富にオプションが用意されており、目的に応じたビルドアップが可能です。

機能例●ラベリング・形状計測・円型分離・位置計測・寸法計測・マルチウインド・２値化・疑似カラー・反転・等輝度・濃度断面・濃度ヒストグラム・濃度射影・輪郭補正・積分・微分・論理演算・細線化・収縮・膨張・骨格化・その他

HIGHSPEED FLEXIBLE IMAGE PROCESSOR

PIP-4000

株式会社 エー・ディー・エス

東京本社 〒113 東京都文京区本郷2丁目10-9(富士ビル2F)
TEL. 03-818-1031 FAX. 03-818-6038

大阪支社 〒559 大阪市住之江区南港東8丁目2-67
TEL. 06-613-5871 FAX. 06-613-5858

パソコン、オフコン、ミニコン
PFU

VOL.1

For R&D

PFUのワークステーション第一弾をお届けします。

デスクトップモデル11/12　小型デスクサイドモデル21/22/23　デスクサイドモデル31/32

製品の開発にあたって、私たちは、ワークステーションは、使う人の目的やニーズに応えるオープンなツールでなければならないと考えました。その思いをかたちにしたVOL.1、UNIXワークステーションPFUΣstation230。研究・開発の効率アップに向けてCPUにはMC68030を搭載、誰もが共通に利用できるプラットホームとなるようUNIX環境、国際標準・業界標準を徹底してサポート。R&Dワークステーションとして、使いやすい環境を備えたツールであると自信を持っています。それには、実は理由があります。いち早くUNIXシステムに取り組んできた実績と経験。さらに、パソコン開発から得た使いやすいMMIを、オフコン開発で培った膨大なノウハウを、スーパーミニコンの圧倒的なパワーを、このワークステーションに結集しているからです。実際にお試しいただいて、その性能を満喫してください。

基本仕様●CPU:68030(25MHz)●FPP:68882(25MHz)●浮動小数点演算アクセラレータ:標準装備(モデル23/32)●OS:UNIX SystemV[*1]+4.2BSD(一部機能)●キャッシュメモリ:32KB(モデル11/12/21/22/31)、64KB(モデル23/32)●メモリ:最大24MB(モデル11/12)、最大32MB(モデル21/22/23)、最大48MB(モデル31/32)●基本ディスク:標準80MB(モデル11)、標準134MB(モデル21)、標準180MB(モデル12)、標準330MB(モデル22/23/31/32)●標準インタフェース:RS-232C、セントロニクスほか●ネットワーク機能:NFS[*2]、Ethernet[*3]、TCP/IP、FNAほか●開発支援ツール:GKS、統合文書処理Augusta、X-window[*4]、Σツールほか●ディスプレイ:19インチモノクロ、16インチカラー、20インチカラー

※1 UNIXオペレーティングシステムはAT&Tベル研究所が開発し、AT&Tがライセンスしています。
※2 NFSはSun Microsystems社の登録商標です。 ※3 Ethernetは富士ゼロックス社の登録商標です。
※4 X-windowはマサチューセッツ工科大学で開発されたソフトウェアです。

UNIXワークステーション
誕生、PFU Σstation230

本広告に掲載の全商品ならびにそれに関連する消耗品等および役務について、ご購入の際、消費税が付加されますのでご承知おき願います。

東京本社/営業部〒105東京都港区西新橋3-25-33(NP御成門ビル)☎(03)438-0311
名古屋営業所/〒460名古屋市中区錦1-19-25(東京生命館)☎(052)202-0871
大阪営業所/〒530大阪市北区堂島1-5-17(堂島グランドビル)☎(06)344-3648
広島営業所/〒730広島市中区基町12-3(朝日生命広島ビル)☎(082)228-3551
福岡営業所/〒812福岡市博多区博多駅前1-5-1(朝日生命福岡ビル)☎(092)472-7056
Uプラザ /〒105東京都港区西新橋3-24-9(飯田ビル)☎(03)434-5446
EPOCプラザ/〒163東京都新宿区西新宿2-4-1(新宿NSビル・OAセンター内)☎(03)342-0280

(資料請求番号　26)

Ver.2.0

パソコンでフルカラーのソリッドモデリング

『マドレールSF』は、円柱、円錐、球、回転体、柱体といったプリミティブのブーリアン演算を行うことにより、複雑な３次元物体を生成するソリッド・モデラーです。モデリングはキーボードまたはプログラムによってインターラクティブに行えます。自動的にプログラムを生成するレコード機能も備えています。

データ表現はB-reps(境界表現)となっています。入力および出力部分は公開されていますので、他のアプリケーションのCSG (Constructive Solid Geometry)形式のデータを『マドレールSF』の入力データとしたり、『マドレールSF』でモデリングしたものを他のワークステーションやミニコンなどでレンダリングしたりすることが可能です。

稜線表示、簡易陰線消去表示、陰線消去表示、半透明なコンスタント・シェーディング表示、レイトレーシング表示といった表示モードがあり、これらのモードを組み合わせた複合表示も可能です。フレームバッファを使用すれば、1677万色を使ったシェーディング表示が行えます。半透明なコンスタント・シェーディング表示は稜線表示と同程度の速度で表示をすることができます。

数値演算プロセッサ8087、80287、80387を使用すれば高速な演算が可能となります。数値演算プロセッサがなくとも高速な演算が可能なプログラムも用意されており、ラップトップ・パソコン上でモデリングすることも可能です。

現在対応しているパソコンは、PC－9801およびPC－286シリーズです。フレームバッファはサピエンス社のスーパーフレームが使用可能です。フレームバッファが実装されていない場合はPC－9801のグラフィックス画面に4096色相当のタイリング表示で出力します。

※東京秋葉原T・ZONEにて『マドレールSF』のデモを行っております。
※『マドレールSF』、『UCG』ともにPC-9801RA、MS-DOSバージョン3.3に対応しています。

●価格／￥118,000
●開発・販売／株式会社　アークブレイン

浅葉肇　作　　鈴木馨　作

Ultra C Graph Ver.1.52

PC-9801、PC-286対応 C言語用超高速 グラフィックス・ライブラリ

■Quick C Ver.1.1、Microsoft C Ver.5.1、4.0、TURBO C Ver.2.0、Lattice C Ver.4.0に対応しています。(各別製品)

■グラフィックス関数は全てアセンブラで記述されており、しかもLIOをコールせずに、GDCやグラフィックスVRAMに直接アクセスしているため、究極のスピードを実現しています。6種類の描画モード、128倍迄の画面ハードコピーなどUCGならではの機能が豊富にあります。高速CADなどのアプリケーション開発用に最適です。

■ANSIの規格に対応したプロトタイプ宣言がしてあるため、返り値と引数の数と型のチェックが厳密に行えます。

※当社に直接お申し込みの方に限り、テキスト画面の拡大ハードコピーが行えるユーティリティのソース・プログラムを差し上げます。

●価格／￥32,000　●開発／株式会社　アークブレイン

※販売店様用に『マドレールSF』、『Ultra C Graph』のデモプログラムを用意しております。

開発スタッフ募集
※新卒者同時募集※見習も可

Archbrain

株式会社 アークブレイン 〒151 東京都渋谷区幡ヶ谷3-20-2 グランドメゾン幡ヶ谷207 TEL.(03)375-8968 FAX.(03)375-8767

■お申し込み・お支払い方法■

『マドレールSF』、『UCG』はパソコン・ショップまたは当社で直接お買い求めいただけます。住所・氏名・電話番号・商品名(UCGではC言語の種類も明記)・個数・ディスクの種類を明記のうえ電話、FAXまたは封書にてお申し込み下さい。お支払いは銀行振込、現金書留または郵便振替にてお願い致します。入金確認後、即日発送いたします。送料は当社で負担いたします。

《振込銀行》
八千代信用金庫　笹塚支店　当座　005-1043492
第一勧業銀行　笹塚支店　当座　161-0115523
三菱銀行　笹塚支店　普通　138-4857406
《振替口座》　東京1-251652

●MS-DOS、Microsoft C、Quick CはMicrosoft社の商標です。●TURBO CはBorland社の商標です。●Lattice CはLattice社の商標です。

CG倶楽部のこれまでの月例会の主なテーマ

87年12月にスタート以来これまでに，コンピュータによるリアルなイメージ生成を中心とする次のようなテーマで月例会を毎月開催しております。

- ◉3次元CGシステムWavefrontの解説とデモ（東京）
- ◉地方放送局におけるCGへの取り組みについて（静岡）
- ◉建築および景観のCGシミュレーションについて（京都）
- ◉大村皓一のCG道場（東京，大阪）
- ◉低価格なパーソナルCGシステムの現状報告（東京）
- ◉シャープにおけるCGへの取り組みの現状（大阪）
- ◉SIGGRAPH'88報告―技術と製品，アニメーション（東京）
- ◉SIGGRAPH'88インターナショナル・ミーティング（アトランタ）
- ◉誰にもできるパソコンによるCGアニメーション（東京）
- ◉名古屋大学横井研究室におけるCG研究とデモ（名古屋）
- ◉管面撮影法による簡易CGアニメーションの作り方（東京）
- ◉ヨーロッパのCG作品上映（東京）
- ◉TBSにおけるCGの放送への取り組み（東京）
- ◉筑波大学芸術工房見学と三井秀樹のデザイン学（筑波）
- ◉ハイテックラボにおけるCG映像と制作の実状（東京）
- ◉リンクスにおける最先端CG映像制作の現状（東京）

会員募集中

コンピュータ・グラフィックス倶楽部（CG倶楽部）は，コンピュータ・グラフィックスがもつさまざまな可能性のなかで，「より豊かな，よりダイナミックな映像表現力」に注目し，その表現力を限りなく追求するための倶楽部です。

▼

活動分野はアートやデザインのほかに，「コンピュータ・グラフィックスによる映像表現力」に関連する分野であれば，シミュレーション，プレゼンテーションなどの幅広い分野を含み，特に分野を限定するものではありません。

●

この倶楽部は営利を目的とするものではありません。また，特定の企業や団体の利益にそって活動するものでもありません。したがって，会員の総意による運営を行います。

■

CG倶楽部は本部事務局を東京に置き，国内外各地に支部（ローカル・グループ）をつくり，国際的な活動を積極的に行います。

1988年10月現在，国内で開設した支部は下記の通りです。

●静岡支部　●名古屋支部　●大阪支部　●福岡支部

［CG倶楽部世話人］

- ◉大村 皓一（大阪学院大学）
- ◉國井 利泰（東京大学）
- ◉中前 栄八郎（広島大学）
- ◉三井 秀樹（筑波大学）
- ◉山本 強（北海道大学）

【規　約】

■主な活動

- ●月例会（毎月開催。ゲスト講演と会員の発表，会員間の懇親などを行います）
- ●特定のテーマごとの部会活動（部会の方針で活動します）
- ●支部ごとの支部活動（支部の方針で活動します）
- ●機関誌の発行（会員間のコミュニケーションや活動案内の掲載を行います）
- ●パソコン通信によるメンバー間のコミュニケーション
- ●年次総会（年一回開催する会員の技術，作品および利用事例などの発表会）
- ●国内，海外の関連団体との交流
- ●各種のイベントの開催

■支　部（ローカル・グループ）　会員5名以上により地域，職場，学校などごとに支部を作り，支部活動を行うことができます。

■部　会（SIG）　会員は特定テーマや分野ごとに部会を作り，部会活動することができます。

■会員の資格　会員になるために特別な資格を必要とすることはありません。

■会　員　会員は個人の正会員を基礎とします。

■会　費　入会金　￥2000（入会時のみ）
年会費　￥2000（毎年4月1日より翌年3月31日まで）

※入会は年度単位とします。4月1日より翌年の3月31日までを年度とします。

※会員には倶楽部機関誌をお送りします。

※年会費には月例会などの各種の活動に参加する費用は含まれていません。

※詳しくはパンフレットを事務局までご請求ください。

お問合せ・お申込先●CG倶楽部事務局／〒101 東京都千代田区神田神保町1-64 神保町協和ビル 図形処理情報センター内
☎03(293)6161　FAX 03(293)6164

明日の産業と豊かな環境を統合する 知的Computer Graphics

CG COMPUTER GRAPHICS OSAKA'89

展示会

- 会期：1989年6月14日(水)～17日(土) 10:00～17:00
- 会場：マイドームおおさか
- 展示内容：CAE/CAD/CAMシステム
  EWS(エンジニアリングワークステーション)
  グラフィックス入出力機器
  各種グラフィックスアプリケーション
  画像処理機器・システム　他
- 特別企画：デスクトップパブリッシング(DTP)コーナー
  (ローエンドDTPシステムと日本語対応ポストスクリプトプリンターを展示実演いたします。)
- 入場料：1,500円

シンポジウム

6月14日(水)		6月15日(木)		6月16日(金)	
10:00～13:00	14:00～17:00	9:30～12:30	14:00～17:00	9:30～12:30	14:00～17:00
オープニングセッション・基調講演・特別講演	A－1 先進CAD/CAMユーザーの事例	A－2 最新CAD/CAMと関連技術の動向	A－3 金型設計・製作のための先端CAD/CAMとユーザーの活用事例	A－4 先進CAEユーザーの事例	A－5 形状入出力の最近動向
	B－1 エンジニアリングワークステーションの基幹技術	B－2 CASEツールの現状と活用具体例	B－3 論理合成ソフトウェアによるシステム設計の自動化	B－4 CGソフトと画像処理(超リアリズムの追求)	B－5 CGハードウェア ---グラフィックス・ワークステーションのアーキテクチャ---
	C－1 医用・生化学の画像処理・先端現場からの実例紹介	C－2 高度消費化時代のCGパートⅠ ---ファッションと心理の追求---	C－3 高度消費化時代のCGパートⅡ ---販売戦略のCG---	C－4 CG活用による過去から未来へ都市と空間の創造	C－5 環境向上に活躍するマッピング技術
	D－1 FA/CIMの経営戦略に果す役割	D－2 FA/CIMの先端構築手法とその活用事例	D－3 マルチメディア情報システムとその具体例		
	T－1 シミュレーション・ビジュアリゼーション技術と活用例	T－2 AI(基礎)知的CGのための基礎技術	T－3 AI(応用)エキスパートシステムの応用事例	T－4 知的ネットワーク構築のための基本技術	T－5 高度ヒューマンインターフェイスをもつ分散処理環境

●詳しい案内書を作成中です。ご希望の方はCG OSAKA事務局までご請求ください。

CGデザインコンテスト ■CG静止画像■

①応募テーマ

A：CGとVisualization
(CG OSAKA'90のメインデザインとして使用可能な作品)

B：フリー
(テーマは特に限定しません)

②賞品

★デザイン大賞 1点：賞金20万円と副賞
★優秀賞 2点：賞金10万円と副賞
★入選 数点：賞品

デザインコンテスト
資料請求券
PIXEL 5

応募要領をお送りいたします。ハガキに資料請求券をお貼りのうえ、CG OSAKA事務局までご郵送ください。

CG OSAKA '89に関してのお問い合せは
CG OSAKA'89 事務局

JMA 社団法人日本能率協会 関西事業本部内
〒541 大阪市中央区安土町2-3-13 大阪国際ビル TEL 06(261)7151

(資料請求番号 29)

1989 5
No.80
1989年5月1日発行　毎月1回1日発行

◎PIXELの購読料等に消費税が加算されます。 消費税の4月1日導入に伴い，PIXELの定期購読，別冊号等に消費税3％が加算されますのでご了承下さい。なお，書籍は消費税を含んだ価格を定価として表示いたします。

地図情報処理技術視察団11月催行 11月12日から15日まで，米国ロサンゼルスで開催される，地図情報処理に関する大規模な国際会議/展示会であるNCGA's Mapping & GIS'89を視察するとともに，米国の先進的なユーザー，ベンダー，研究機関などを訪問する海外技術視察団を派遣いたします。

●表紙の解説

「GERM」

実はこの絵はデータの量がパソコンの限界をかなりオーバーしていて，ビルを数個ずつのブロックに分け，8回のレンダリングを重ねてようやく出来上り，という手順をふんでいます。窓はリフレクトのきいたテクスチャマッピングとバンプマッピングを重ねて張り付けました。柔らかくボケてしまうのがいやだったのでアンチエイリアシングは行っていません。

僕は作品を作るにあたって，まずイメージを先に立たせて，方法は後からくふうして追いつかせる事で自分と作品がより強い関係を結ぶ事が出来ると思っています。

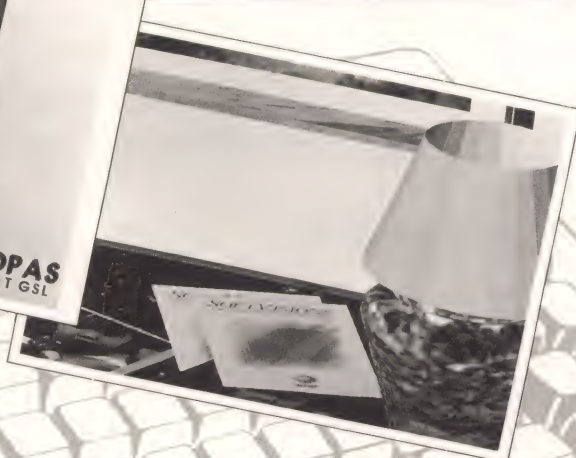
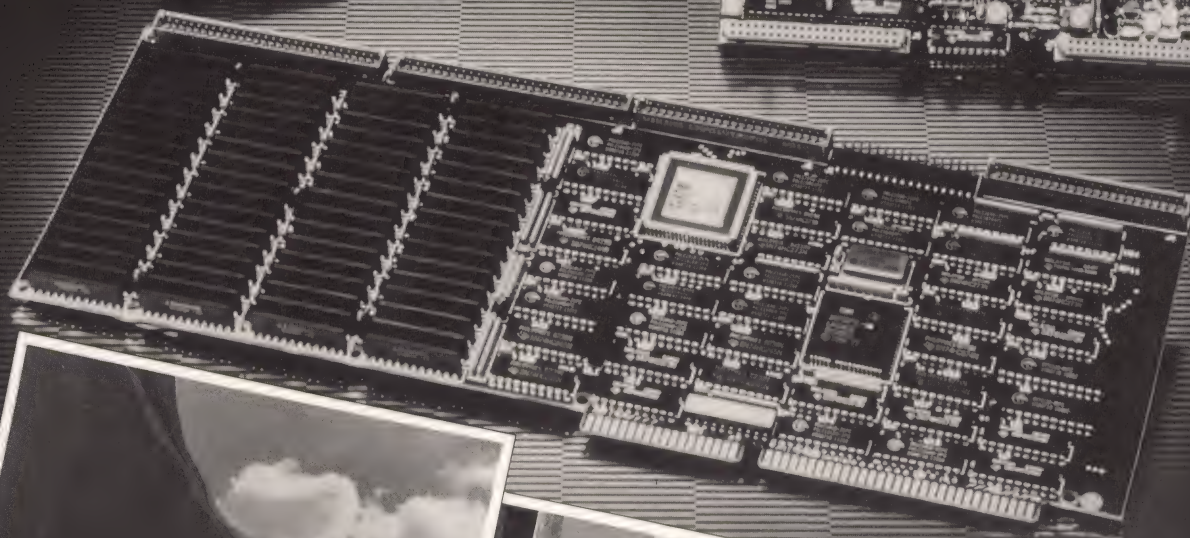
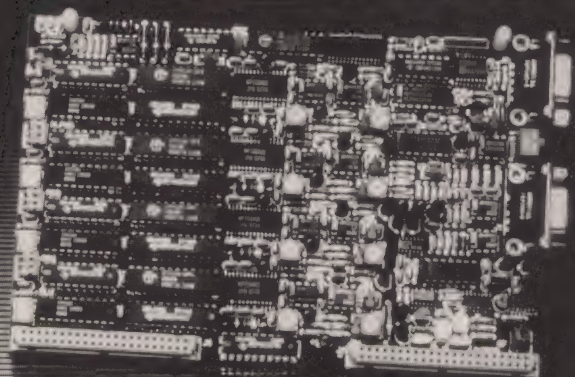
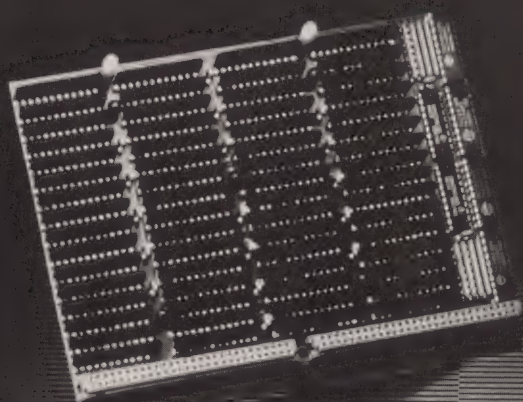
僕がつくる「絵」に，見る人の心を動かす力が宿ってくれる事を望んでやみません。

ハード　NEC　PC 98 VX 21
ソフト　パーソナルリンクス　TRACY
須藤牧人

ビデオで見るCG最前線II(日本語版) 今春販売開始し，大好評いただいた「ビデオで見るCG最前線」(ACM/SIGGRAPH監修)の日本語版が出来ました。内容も英語版未収録のものも加わり，さらに充実。CGの最新情報は「……II」日本語版で！　詳しくはお知らせページ(36ページ)をご覧ください。

PIXEL 編集スタッフ募集 PIXEL 編集部ではPIXEL の誌面充実のため，編集スタッフを募集します。コンピュータの知識および経験があり，編集に意欲的な方を求めます。詳しくはお知らせページ(40ページ)をご覧ください。

TRUEVISION INC.

AT&T GSL
RIO

TOPAS
AT&T GSL

TRUEVISION社は、U.S.A.の
コンピュータグラフィックの分野で代表的な会社です。
その製品は TARGA、ATVista、MuVistaの3シリーズがあります。コンピュータは IBM、AT コンパチ機及び、MAC が使用できます。最高の機能のものは、4MBのVRAMバッファと1,024×1,024 pixel×32bitのアドレス空間を有しております。

ソフトは各社から供給されております。例えば TOPASは3Dモデルおよびアニメーションのソフトで、また RIO は高品質のイメージとデザインを組み合わせるソフトです。T/V CAD はAutoCADとビデオイメージをオーバレイすることができます。その他建築、インテリア、グラフィック、広告、化粧、整形、生地、織物、ファッション用のデザインソフトがあります。日本語のフォントも使用できます。

私共はソフトとハードをいろいろな組合せで提供します。

TRUEVISION INC.

日本総代理店

〒108 港区芝4-5-12 三田ハイツ2F
☎(03)455-3460(代) FAX(03)457-7970
(JR田町駅三田口歩5分 営業時間9:00〜18:00 日曜、祭日、第2・3土曜定休 土曜12:00迄)
✣全ての商品は日本信販の分割払い、リースを御利用できます。

(資料請求番号 30)

CAD/CAMアルファ

CAD/CAMアルファ・シリーズは，CAD/CAM/CAEの基礎技術，最新の話題，運用のノウハウなどについて，いま最も必要としている記事を満載しています。

各号は大きなテーマごとに編集され，初心者から経験者まで，ユーザーからシステム開発者まで，どの層の方にもお読みいただけます。

No. 1

大特集 どうなる，これからのCAD/CAMとCAE!

- 最近のパーソナル CAD とエンジニアリング・ワークステーション
- エンジニアリング・ワークステーションとソフトウエア
- パーソナル CAD と分散処理
- 画像処理の製造業への応用

一般

- やさしい CAD/CAM 入門の入門
- わが国における CAD/CAM 導入の現状
- CAD/CAM システムの研究

No. 2

大特集 いま最も注目されているパソコン CAD の実力と使い方

ユーザー事例

- ミサワ CAD システム（建築）
- 岡村製作所（機械設計）
- リコー（LSI 設計）
- 大信工業，重野製作所（金型）
- プラスチック金型設計

新領域に臨むパソコン CAD

- ハンドヘルド CAD
- パソコンに接続する小型モデリング・マシン

シリーズ・アプリケーション キャノン　東芝　日立　ブリヂストン

A4変型判／定価各号1,545円(本体1,500円＋税45円)

PIXEL 別冊

アプリケーション

コンピュータ・グラフィックス ユーザー事例集

製造業における CAD／CAM を中心に，アニメーション，建築・土木，構造解析，マッピング，リモートセンシング，アパレル などの各分野における代表的ユーザーを1冊に収録しました。これから導入を検討されている企業，機能アップを考えている企業にとって見逃がせない1冊 となります。

- 東洋工業におけるCAD/CAM
- 自動車部品へのCAD/CAMの適用
- 航空機内装品生産におけるCAD/CAMシステムの利用
- 石川島播磨重工業での船殻CAD/CAMの現状
- 日本建鉄の金型加工におけるCAD/CAMシステムの適用事例
- 立石電機におけるCAD-CADAMシステムの紹介
- CAD一貫システムへの取組みとその事例
- 冷間コイルばね成型へのCAE導入
- 光学設計分野におけるCAD/CAM
- 有限要素法解析における図形処理の一例
- コンピュータ・グラフィックスの分散統合処理へのアプローチ
- プラント機械設計用CADシステムの開発と適用
- 電設メーカーにおけるCAD/CAM適用例
- 集積回路設計におけるコンピュータ・グラフィックスの利用
- 電子回路基板におけるCAD/CAMの適用
- 日立電子におけるCAD/CAMの適用
- プリント配線基板の設計・製造へのCAD/CAMの導入事例 他

定価4,635円(本体4,500円＋税135円)

グラフィック・システムを効率的に使いこなすためには，他社での事例を研究することがきわめて重要です。本書は導入の検討，システムの設計と選択，標準化などの受入れの準備，実際の運用，評価と今後の展望——などについて，さまざまな業種・分野からの事例報告をまとめました。

お申込みは巻末の葉書をご利用下さい。

※お申込みは巻末のはがきをご利用ください。

編集◉発行／図形処理情報センター
〒101 東京都千代田区神田神保町1-64神保町協和ビル
☎03(293)6161　Fax03(293)6164

バックナンバー

87年 1 月号(No.52)

大型企画：コンピュータ・グラフィックス小史(前編)■特集1 CAD/CAM, CGのベンチャー企業□座談会-CAD/CAM, CGの分野では高度専門技術を持った少人数精鋭企業の果たす役割がますます大きくなってくる□インタビュー-娯楽分野の映画制作からCG関連全体にビジネスを拡大したい□CGの専門学校として新しい教育のあり方を探っていきたい□本当に使って役に立つCAD/CAMとは何かを常に考えている■大型企画：パーソナル・ソリッド・モデリング「マドレール」の機能と使い方■特集2：SIGGRAPH'86の注目論文抄訳□自然現象の表示□レンダリングの高速化技法□光源モデル■ほか

87年 2 月号(No.53)

特集1：パーソナルコンピュータによる新しいアイデアのCG■パーソナルコンピュータとRAMボードによるリアルタイム・アニメーション■パーソナルコンピュータと8㎜カメラによるCGアニメーションの制作■パーソナルコンピュータによるレイ・トレーシングの特殊表現■パーソナルコンピュータによる人体モデルの生成とその応用□特集2：ワークステーションで利用できるCAD/CAM, CAEシステムとソフトウエア□企画：プロのCG映像ができるまで□一般：SIGGRAPH'86の注目論文抄訳■CAEによる建築設備設計システム-APEC照明計画編■ほか

87年 3 月号(No.54)

特集1：期待される地図情報処理①──最新システムとその使い方(前)□道路,上水道,下水道,資産税を中心とした総合的な都市情報システム□解析機能に優れた地理情報システムARC/INFO□総合データベースを実現するコンピュータ・マッピングIIS-MAP□都市情報システムの構築に新しい概念でこたえるINS-SPACER□富士地図情報処理システムFAMOS□自由度の高いマッピング・ツール INFORMAPII□INTERGRAPHシステム■特集2：CAD/CAM,CGのベンチャー企業(後)海外ベンチャー企業の現状■コンピュータ・グラフィック小史■カラー企画:CG年賀状■ワークステーションで利用できるCAD/CAM, CAEシステムとソフトウェア(後)■ほか

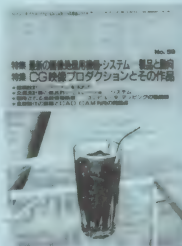
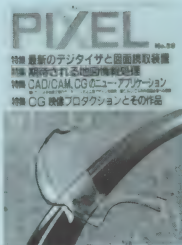
87年 4 月号(No.55)

特集1：CG映像プロダクションとその作品①□トーヨーリンクスのCGプロダクションとその実際□CG映像の多様化を目指すGCGC□映像制作におけるCGテクニックのクロスオーバーはますます進む□コンピュータ・アニメーション・システムANTICS■特集2：期待される地図情報処理②──最新システムとその使い方(後)■EWSベース統合マッピング・システムCADD station■多様化するマッピングニーズにこたえ体系化を図った日立地図情報システム■地域情報・施設情報を総合的に管理する■ラスター・イメージ・パソコン・システムGRIS■総合建設コンサルタントにおけるコンピュータ・マッピングOHBA GIS■ほか

87年 5 月号(No.56)

特集1：CAD/CAM, CGのニュー・アプリケーション□設計者が簡単に使えるFEM解析システムCAD-FEM連動システム□アプリコンBRAVO!による山留・桟橋設計システム□EWS支援ツールとしてのパーソナルコンピュータ□GCADシステムによる自動設計の試みと外部インタフェース■特集2：期待される地理情報処理③──最新システムとその使い方■コンピュータ・マッピングの最前線■土地管理から計画策定支援まで地理情報システムWING■都市情報システムの構築を支援する自治体地図システムARISTOWN■エリア・マーケティングのためのコンピュータ・マッピング・システムKITE■地図データベース構築における図面自動処理システムの動向■ほか

87年 6 月号(No.57)

シリーズ企画：ユーザーのためのCAD/CAM, CG, 画像処理用機器・システム②〔グラフィック・ディスプレイ〕□グラフで見る最新グラフィック・ディスプレイ調査結果□グラフィック・ディスプレイの最新動向□グラフィック・ディスプレイ製品ガイド■特集：期待される地図情報処理④□コンピュータ・マッピングの最前線②□なぜ森林にコンピュータ・マッピングか─森林管理データマップシステムRobin Hood□地理情報解析システムGEO log■シリーズ：CAD/CAM, CGのニュー・アプリケーション□IGESを利用した電子カタログサービス□パッケージ設計支援システム■ほか

87年 7 月号(No.58)

特集：期待される地図情報処理⑤□コンピュータ・マッピングの最前線③□神奈川県における都市情報システム□LAMS誕生の背景について□明電地図情報利用システムEMAP■シリーズ企画:ユーザーのためのCAD/CAM, CG,画像処理用機器・システム③〔デジタイザと図形読取装置〕□2次元マニュアル・デジタイザの位置付けと原理概説□図面認識技術とオートデジタイザ□最新のデジタイザ製品ガイド■シリーズ:CAD/CAM,CGのニュー・アプリケーション□CGによる伝統文様データベース化と工芸デザインの開発□CAD/CAMの関連企業への展開■CGプロダクションとその作品③■ほか

87年 8 月号(No.59)

シリーズ特集：CG映像プロダクションとその作品⑤20社■シリーズ企画：ユーザーのためのCAD/CAM, CG,画像処理用機器・システム④〔画像処理用機器・システム〕□Dr.SPIDERのワンポイント・アドバイス□最新の画像処理用機器・システム□画像処理用機器製品ガイド■シリーズ：CAD/CAM, CGのニュー・アプリケーション□建築設計にリアリティを求めて□交通流計画の総合的シミュレーション・システムTRAFFIC PLAN■大型企画：コンピュータ・グラフィックス小史(後編)■シリーズ：金型専業メーカーにおける実践的CAD/CAM活用事例Ⅰ□金型製作の課題とCAD/CAM利用の問題点■シリーズ企画：期待される地図情報処理■ほか

87年 9 月号(No.60)

シリーズ企画：ユーザーのためのCAD/CAM,CG,画像処理機器・システム⑤〔ハードコピー機器・装置〕□最新のCAD/CAM用プロッタ□静電プロッタ概論□サーマル・プリンタ/プロッタの概論□最近のカラーインクジェット・プリンタの動向□デジタル方式フィルムレコーダ概論□最新のハードコピー機器・装置□ハードコピー機器・装置製品ガイド■シリーズ特集：CG映像プロダクションとその作品⑥放送局におけるCG画像(前)□NHK□日本テレビ□読売テレビ□フジテレビ■CG：ユタ大学で開発された新モデリング・システム alpha-1■ほか

87年 10 月号(No.61)

SIGGRAPH'87報告：スーパーコンピュータ級のグラフィック・エンジンとストーリーをもったCGアニメーション□Pixarを中心に全体がレベルアップ─「オムニバス以後」初のSIGGRAPH'87フィルム&ビデオ・ショー・レポート■シリーズ特集：CG映像プロダクションとその作品⑦放送局におけるCG映像(後)□TBS□朝日放送□テレビ朝日□毎日放送■シリーズ企画：ユーザーのためのCAD/CAM, CG,画像処理用機器・システム⑥CG用ハードウエアとソフトウエア□最新のPCグラフィックス・システムとその選び方□アーチストのための3次元CGテクニック一覧□3次元CGアニメーションの制作過程とシステム■ほか

87年11月号(No.62)

特集：最新グラフィック・エンジンとレンダリング・システム□SIGGRAPH '87ハイライト■シリーズ企画：ユーザーのためのCAD/CAM, CG, 画像処理用機器・システム⑦〔専用ボード, LSI, スキャナとカメラ〕□CG専用ボード, 専用LSI概論□RGS色分解とカラースキャナ□固体撮像カメラの最新動向□ボード, LSI, スキャナ, 固体撮像カメラの製品ガイド■企画：ドライビング・シミュレータ□インタビュー レジナルド・ウェルズ氏□高忠実度ドライビング・シミュレータ■シリーズ:CAD/CAM, CGのニュー・アプリケーション□先染織物シミュレーション・システムTEX-SIM■ほか

87年12月号(No.63)

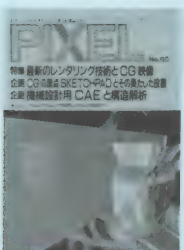
特集:NICOGRAPH'87ハイライト──最新製品と注目論文□複合幾何モデラーの開発□メカトロニクス製品開発におけるCAD/CAEの適用について□等密度テクスチャ・マッピング□ほか■企画:今, どんな製品が売れているか□汎用および機械系CAD/CAM/CAEとグラフィックス・ディスプレイ□ベンダー担当者に聞く■シリーズ：金型専業メーカーにおける実践的CAD/CAM活用事例V□精密ダイカスト金型におけるCAD/CAM/CATの活用と展開■企画:PIXEL記事一覧1987年1月～12月■国際映像ソフトフェア'87─CG部門受賞作品■最新のレンダリングとアニメ技法■ほか

バックナンバー

88年 1 月号(No.64)

特集：コンピュータ映像のいろいろ□フライト・シミュレータにおけるコンピュータ・グラフィックス□展示用特殊映像の種類と歴史□D・CAD□建築設計事務所におけるCGの利用□CGを利用した景観シミュレーション□高品位フォント概論□医療画像処理にみるCGの可能性□電波天文学における画像処理□リモートセンシングの動向□コンピュータ・グラフィックスによるロボット・シミュレーション□ほか■シリーズ企画□ユーザーのためのCAD/CAM/CAEシステム入門［第1回機械設計用CAD/CAMシステム（前編）］■ほか

88年 2 月号(No.65)

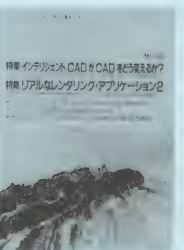
特集：日本のCAD/CAM，CG研究□光と影──その表示法と問題点□商品企画におけるイメージファイリングシステムの利用□ブラックホールのCG□CGのための樹木の手続き的形状定義法について□ほか■シリーズ企画：ユーザーのためのCAD/CAM/CAEシステム入門［第2回電子・電気回路設計用CAD/CAM/CAEシステム］□ASICのいろいろとつくり方──ゲートアレイの開発設計手順──□最新のLSI-CAD技術□ユーザーのためのプリント配線板設計入門□最近のプリント基板におけるCAD/CAMとCIMについて■ほか

88年 3 月号(No.66)

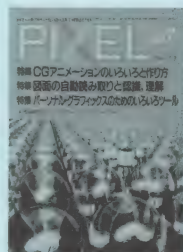
特集：最新のレンダリング技術とCG映像□CG映像制作に使用されるさまざまな表現手法──マッピングを中心にして□トーヨーリンクスにおけるCG映像技術──アルゴリズムとその技術解説□最近のレンダリング・テクニックとアルゴリズム□拡張Zバッファ・アルゴリズム■企画：コンピュータ・グラフィックスの原点SKETCHPADとその果たした役割□アイヴァン・サザランド博士とスケッチパッド・システム□ほか■シリーズ企画：ユーザーのためのCAD／CAM／CAEシステム入門［第3回機械設計用CAEと構造解析■ほか

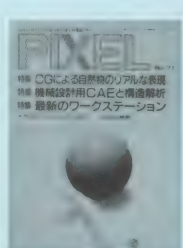
88年 4 月号(No.67)

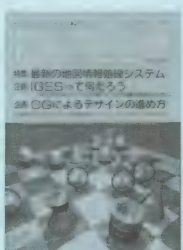
特集：CAD/CAM，CG，画像処理を支える日本の研究最前線(前)□CSGとB-Repsの二重構造ソリッド・モデルと図形処理□ユーザー・フレンドリなシミュレーション技法□Visual Computerにおける要素技術とその応用□CAD研究のブレークスルーを求めて□論理ならびに実装設計の支援システムの研究□VLSIのCADに関する理論的・実用的展開□東京工業大学像情報工学研究施設における画像処理の研究動向□ほか■シリーズ企画□ユーザーのためのCAD/CAM/CAEシステム入門［第4回建築用CADシステム］■ほか

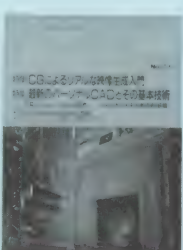
88年 5 月号(No.68)

特集1：リアルなレンダリング・アプリケーション(1)□インテリア・シミュレーションにおけるレンダリング応用□ショップデザインにおけるCADとCGの活用□デザイン業務におけるレンダリング・システムの応用■特集2：CAD/CAM，CG，画像処理を支える日本の研究最前線(後)□実体モデルとその応用□パーフェクト・ソリッド・モデルを求めて□形状モデルを利用した画像処理□知的CADへ向けて□東京大学生産技術研究所高木研究室における画像処理の研究□幾何拘束と幾何推論に基づくCAD/CAMシステム■ほか

88年 6 月号(No.69)

特集1：インテリジェントCADは現在のCADをどう変えるか？　□［1部］インテリジェントCADの第1段階は概念設計に対する支援であり，この5年間のうちに大きな展開が期待される□［2部］座談会□［3部］AI-CADにおける技法とその問題点■特集2：リアルなレンダリング・アプリケーション(2)□自動車デザインにおけるレンダリング・システムの応用□油層シミュレーションにおけるレンダリング技術の応用□眼科臨床におけるレンダリング・システムの応用■新連載□Cプログラミングを用いたレンダリング・ソフトの実践シリーズ(1)

88年 7 月号(No.70)

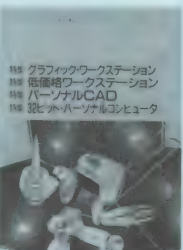
特集1：CGアニメーションのいろいろと作り方□1インチVTRによるアニメーションの作り方□アニメーション・テクニックのCGへの応用□建築・都市のシミュレーション・アニメの作り方□アニメーションのための管面撮影入門■特集2：図面の自動読取りと認識，理解□図面自動入力技術の基礎と動向□図面の自動読取り用イメージ・スキャナ□ワークステーションに適した新しい図面読取り技術■特集3：誰にもできるパーソナル・グラフィックスのためのいろいろツール□パーソナル・グラフィックス入門■ほか

88年 8 月号(No.71)

特集：CGによる自然物のリアルな表現□ランダム・フラクタルを用いた木目と大理石の表現技術□等濃度を用いた雲画像の生成技法□フラクタルによる山と岩石，鋳造物の微細形状表現□水滴衝突の際の流体解析のポスト処理グラフィックスとレンダリング■機械設計用CAEと構造解析□メカニカルCAE（MCAE）とその動向□プリ／ポスト・プロセッサの最新技術動向□最新の運動・機構解析の技術動向■最新のワークステーション□最新のワークステーション□最新のエンジニアリング・ワークステーション製品ガイド□ほか

88年 9 月号(No.72)

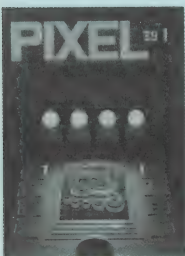
特集：最新の地図情報処理システム□最新の地図情報処理システム／製品ガイド■IGESって何だろう──異なるCADシステム間のデータ交換を進めるために■コンピュータ・グラフィックスによるデザインの進め方□何を描くか，新たな表現手段CG■実用化に向かいつつある最新のレイ・トレーシング・ソフトウエア■パーソナルCADデータ交換のための中間ファイルPCESが10月を目途に製品化■製品モデルの交換仕様STEPの動向─いつIGESが不要になるのか？■コンピュータによる日本語文字の自動デザイン生成法■ほか

88年10月号(No.73)

特集1：CGによるリアルな映像生成入門□スキャンライン・アルゴリズム□Zバッファ，Aバッファのアルゴリズム□レイ・トレーシングを実現するための理論と方法□さまざまなマッピングのアルゴリズム□質感表現のアルゴリズム■特集2：最新のパーソナルCADとその基本技術□パーソナルCADの現状と展望□導入□CADとCG□ワークステーションによる3次元パーソナルCAD/など■SIGGRAPH'88──CG実用化へのスタート■レイ・トレーシングをどう使うか？■CADシステムの新しい将来像「属性モデリング」とは何か■ほか

88年11月号(No.74)

特集1：SIGGRAPH'88とCGの最新動向□レンダリングとアニメーションの実用化□最新のレンダリング／アニメーション技術□ライティング・モデルの研究について□フィルム／ビデオ・ショーおよびアート・ショー・レポート□The Render Man Interfaceの概要□機器展にみるスーパーEWS□新製品ハイライトなど■特集2：図面読取リシステムとその機能□図面自動読取り装置の現状と動向□最新の製品ガイド■最新のインテリジェントCADの製品動向■大手ゲームメーカーのナムコがCG映像ビジネスに参入■ほか

88年12月号 (No.75)

特集1：グラフィック・ワークステーション□最新のグラフィック・ワークステーションとは□グラフィック・ワークステーションの処理能力■特集2：低価格ワークステーション□普及に拍車をかける低価格エンジニアリング・ワークステーション□製品一覧■特集3：パーソナルCAD□機械設計におけるパーソナルCADの評価と導入□パース専用ソフトの導入と運用□パーソナルコンピュータによるシェーディング・プログラム□製品ガイドなど■特集4：32ビット・パーソナルコンピュータ□32ビット・パーソナルコンピュータの最新動向など■ほか

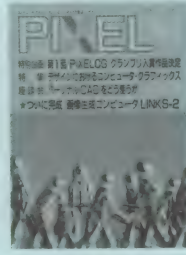
バックナンバー

89年1月号(No.76)

特集：おもしろくなるCG,CAD／CAM, 画像処理□CGプロダクションでの制作の実際□イラストレーションとしてのCG□ランを描く□絵画の手法とCG□グラフィック用コンピュータには何が最適か―SWSから32ビットPCまで□世界のCGプロダクション□CGプロダクションの概要と展望□プリプレスと印刷における電子化□メディア・アート・テクノロジ□Render ManがこれからのCGをアクセレートする□パブリック・ドメインのCGシステムとアマチュアCG活動の動向■ほか

89年2月号(No.77)

特集1：新しいCG表現を可能にするメタボールとその使い方□メタボールと私□メタボール実験プログラム■特集2：3次元機械設計用CAD／CAM―最新技術動向とユーザーにおける使い方□3次元設計用CAD概論□3次元機械設計用CAD／CAM／CAEユーザー調査■特集3：CADシステムのカスタム化□特別寄稿：ここまできたリアリズム□トレンド：いまこそベンチャー企業がカスタムICを使ってGWSをつくる時代だ■ほか

89年3月号(No.78)特大号

大特集：いま使えるCGシステム全調査―ユーザーのための技術と使い方―□いま使えるCGシステム一覧□3次元システムの機能とその使い方□CGアニメーション概論□パーソナルCGシステムを使って一電脳弥七の作り方―□ビデオオリエンテッドなCGシステム～ファインパック□GRAPHIC SYNTHESIZER～彩crone PIXAR用アプリケーション～Chap Reyes□トータルライティング・シミュレーション・システム～TBT□レイ・トレーシング～RAY―TREK CGアニメーション3Dシステム～アートスター■ほか

89年4月号(No.79)

特集：デザインにおけるCG□家具意匠設計におけるデザインシミュレーション□業務用家具分野でのパソコンCGの活用□九谷焼におけるコンピュータデザイン事始め□先染織物におけるCGデザイン・シミュレーション■特別企画：第1回PIXEL□CGグランプリ入賞作品決定□経過報告□入賞作品□グランプリ講評□コンピュータ・グラフィックスとコンピュータ・アート□私のCG界隈■座談会□パーソナルCADをどう使うか□ついに完成画像生成コンピュータLINKS―2□ハードウエアについて□ソフトウエアについて■ほか

グラフィック・デザインにおけるCGのあり方

具体的な制作過程を通して，グラフィック・デザインの中のCGの占める位置を，デザイン事務所の立場から考える。 本文p.p.75〜79

㈱エースデザイン　伊藤 雅俊

図1　ポスター
協力　日本IBM株式会社

図2　林檎の木
図3　永遠に流れる時間

図2

図3

東洋美術学校　杉山 久志

本文p.p.80〜85

コンピュータを道具として自由に使いこなし，感覚的に操作できる。デザイン，製版の流れまで変わってしまう。
これは革命である！

図３　作品例

グラフィック・ペイントボックス

グラフィック・ペイントボックス

グラフィック・ペイントボックス

ビデオペイントボックス＋ミラージュ

ビデオペイントボックス＋ミラージュ

ビデオペイントボックス＋ミラージュ

パソコンによるプリ・プレス処理

昭和製版(株)　行木 修

今やパソコンはミニコンの機能を超え，不可能といわれていた印刷製版のプリ・プレス処理が可能となった。実際にシステムを使用しての現場からの報告をする。

本文p.p.90〜95

図5

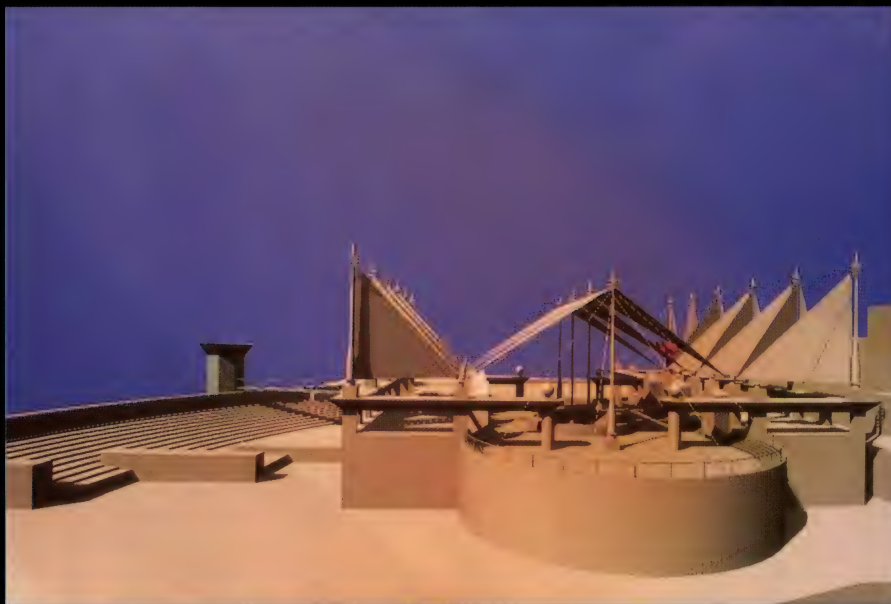
図6

図7

建築設計事務所とCAD

㈱現代建築研究所　　中山 信三

建築設計事務所のCAD利用は，事務所規模や設計業務内容によって多様化しており，今後5年間でCADコンサルなどの新しい形態が生まれようとしている。

本文p.p.97〜103

坂本龍馬記念館構想設計競技入選案

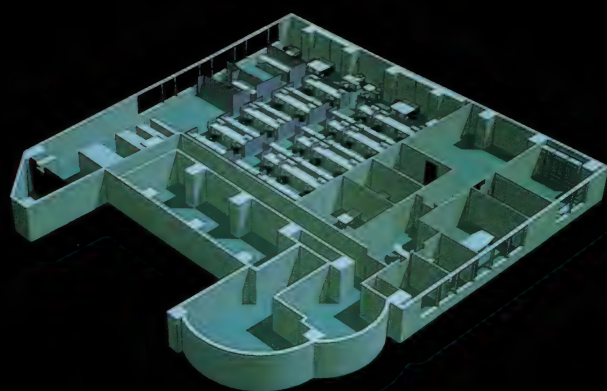



図9　現代建築研究所　事務所レイアウトパース

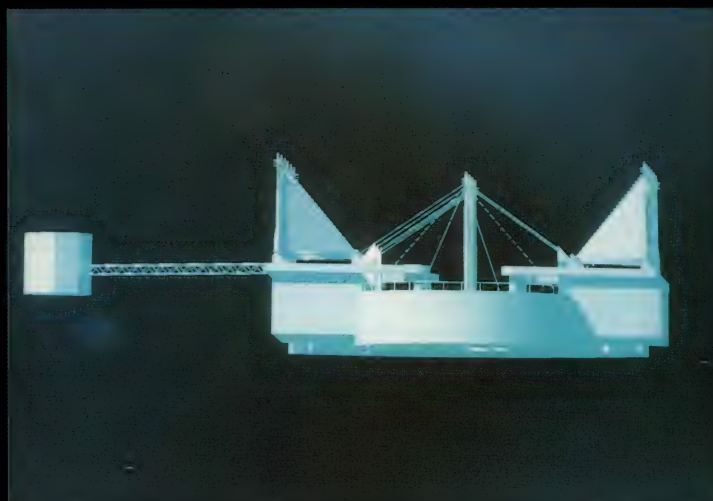



図10　坂本龍馬記念館　デザインシミュレーション段階

図11　坂本龍馬記念館　構想設計競技入選案

図12　高速道路休憩施設　アトリウム内観パース

図 13　高速道路休憩施設　正面外観パース

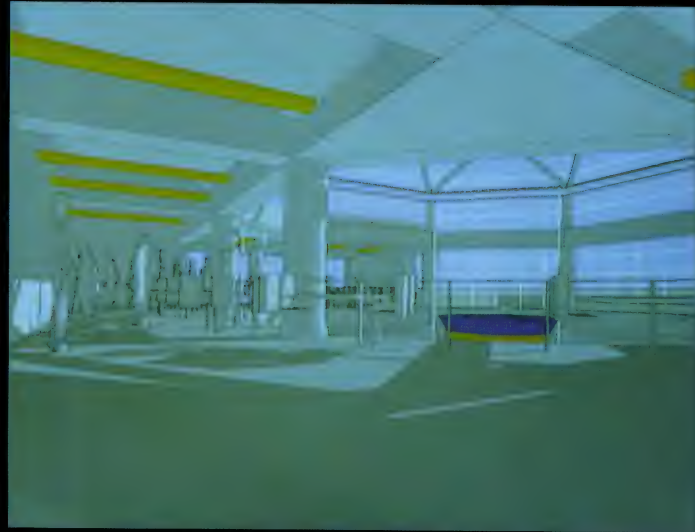



図 14　高速道路休憩施設　レストラン内観パース

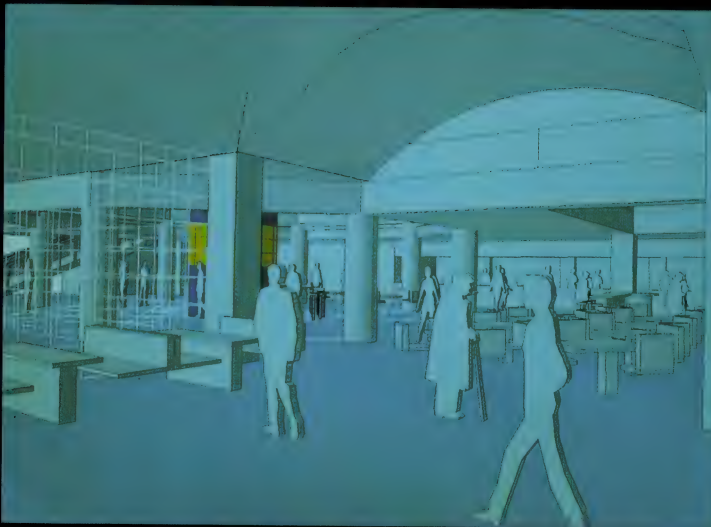



図 15　高速道路休憩施設

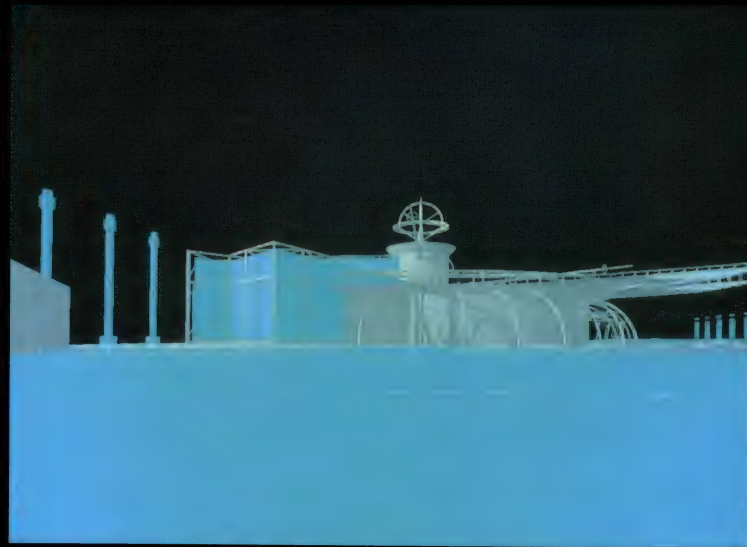



図 16　点光源のシミュレーション

図 17　点光源のシミュレーション　提供：ARC ヤマギワ㈱

図 18　シェーディングのシミュレーション事例　提供：アラップ・アソシェーツ

照明計画とビジュアルシミュレーション

東芝ライテック㈱　菊地 壮一・鹿倉 智明・植田 慶幸

図3　テニスコート照明シミュレーション
—上方光束のない照明器具を用いた場合—

図4　テニスコート照明シミュレーション
—上方光束のある照明器具を用いた場合—

図7　街路照明シミュレーション(1)

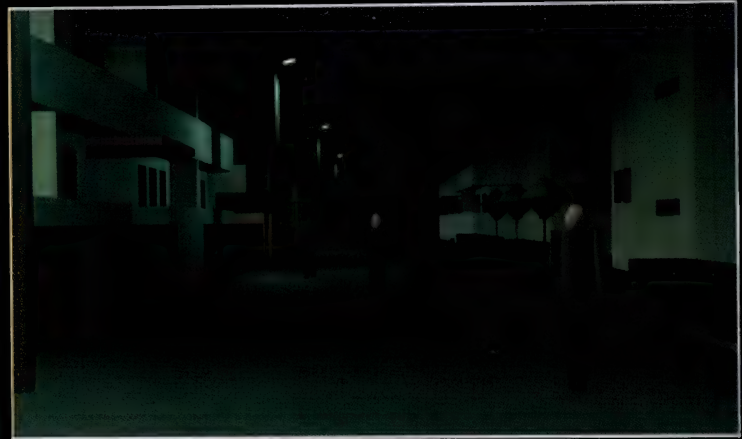



図8　街路照明シミュレーション(2)

当社の照明計画支援CGシステムとその利用事例を紹介し，物理的な裏づけに基づいた照明環境の可視化にCGを利用する効果と問題点，課題について述べる。

本文p.p.104〜107

図9
街路照明シミュレーション(3)
—道路部に高圧ナトリウムランプを使用した場合—

図10　室内における間接照明シミュレーション

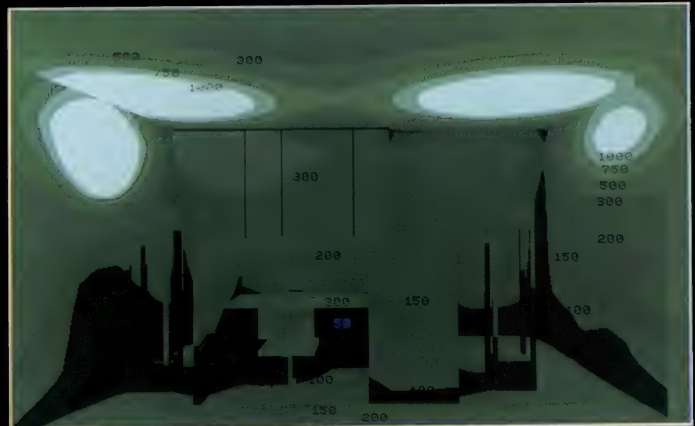



図11　室内における間接照明シミュレーション —照度分布—

大成建設におけるCAEの一例、自然力に対する構造物の応答性状をさぐる

自然力に対する構造物の応答性状を把握することにより構造物の合理的設計ならびに快適な居住空間の創造をめざす研究とCAEのかかわりの一端を紹介する。 本文p.p.120〜124

大成建設(株)　山田 正明

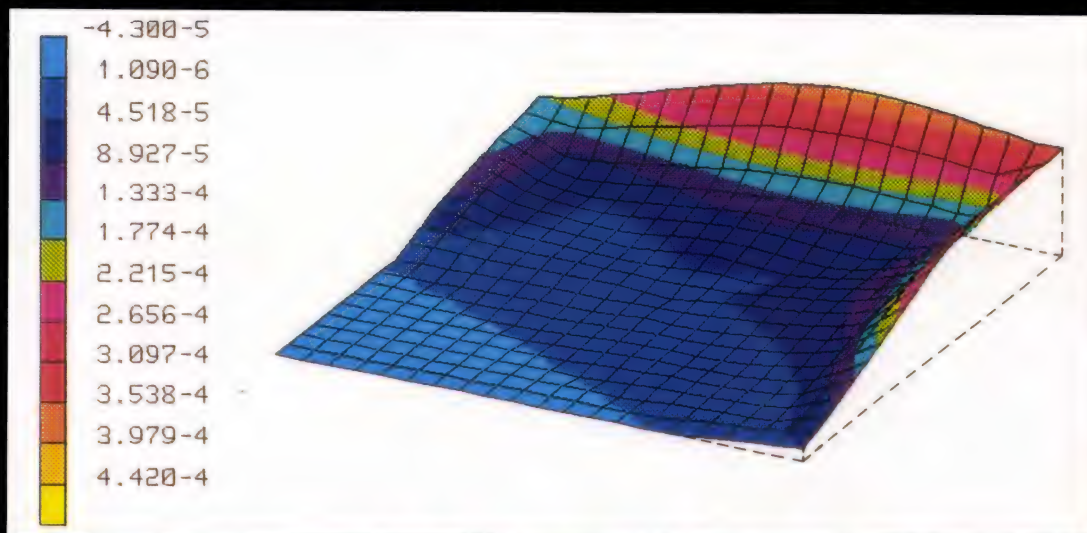



図8　剥離面と地反力分布

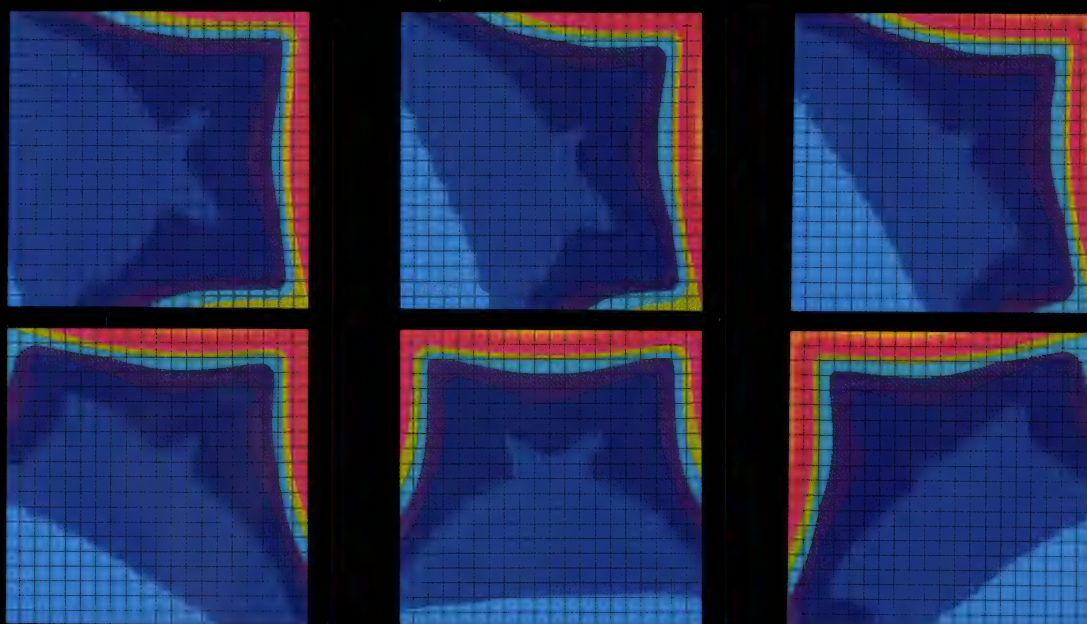



図9　剥離状況と地反力分布の時間的変化

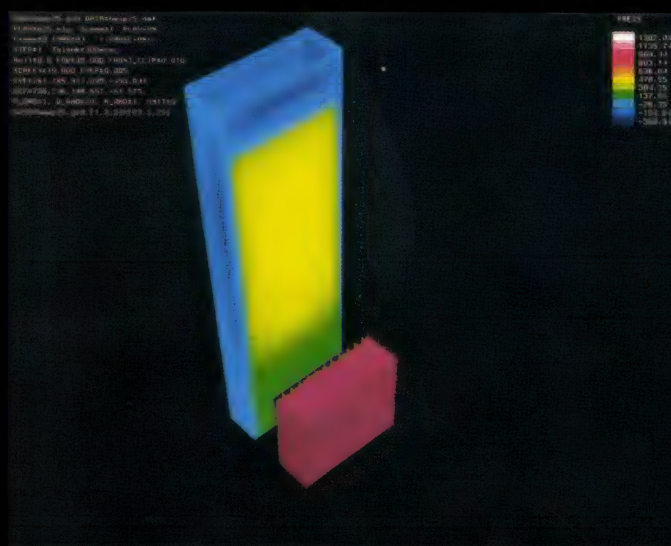



図13　高層構造物に働く風圧分布

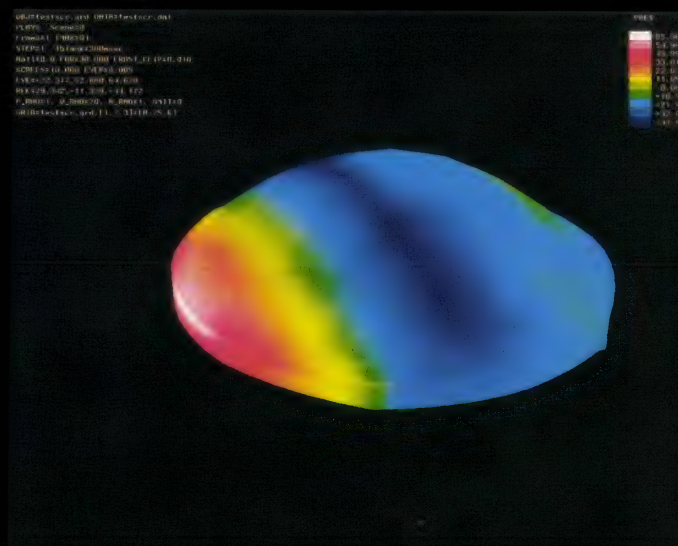



図14　膜構造物に働く風圧分布

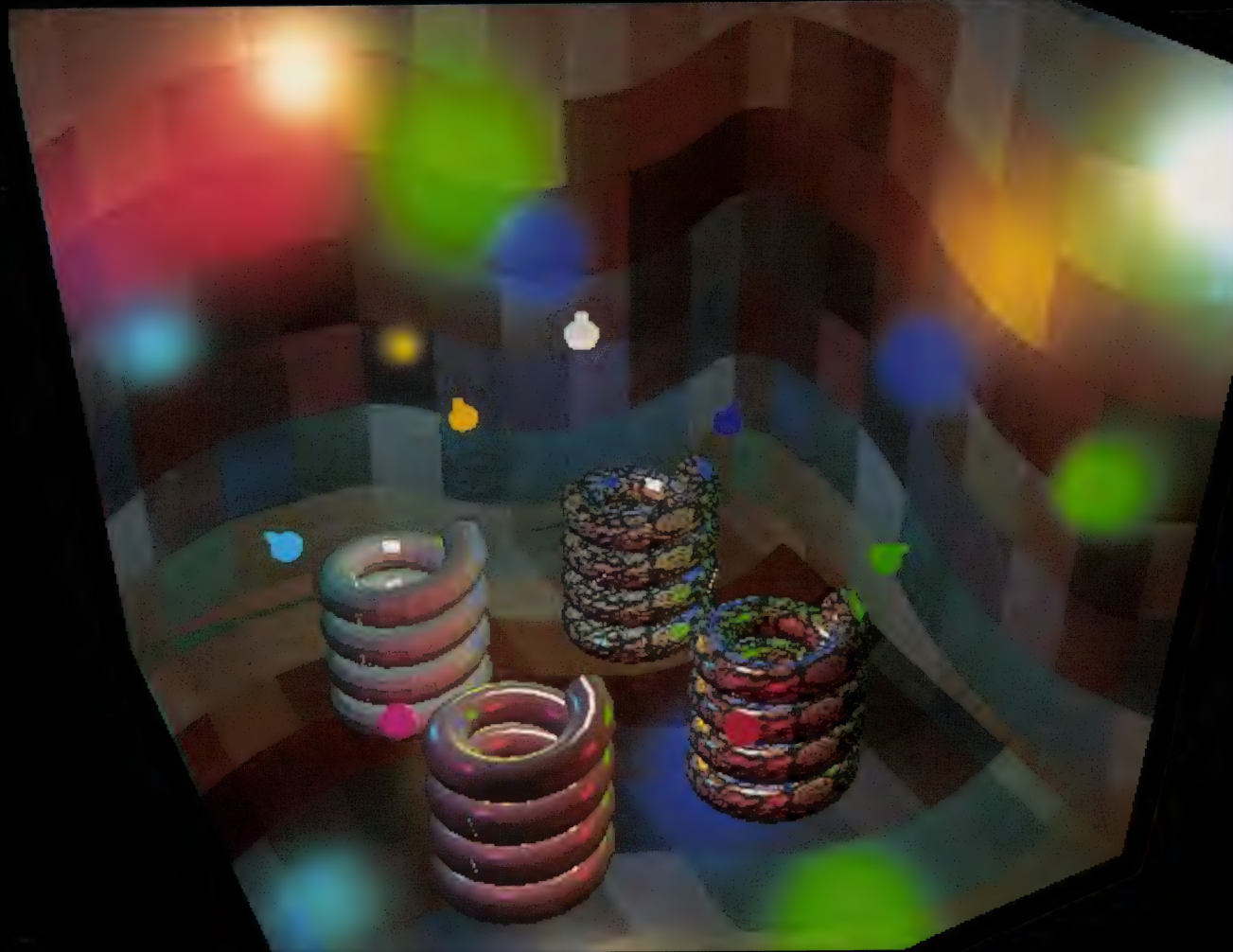
Cプログラミングを用いたレンダリング・ソフトの実践シリーズ（最終回）

出渕 亮一朗

本文p.p.154～164

今回はこのシリーズの最終回として，●簡単なプリミティブ作成プログラムの紹介　●ツリー構造についての補足説明　●debutを使ったテスト・レンダリングと作品例，およびその制作の実際　●debutプログラムの発展 について順次説明していきたいと思う。

▲図5　"Cat & Dog"　ハイテックラボ・ジャパン制作

▼図6　"Coils"スムースシューティング・マッピング・点光源のテスト。

プラント計画及び建設設計へのコンピュータグラフィックスの応用

㈱日立製作所　日立工場　原子プラント設計グループ
主任技師　好永 俊昭・中嶋 明
吉田 直人・吉田 美樹

原子力発電プラントは膨大な量の機器や配管などで複雑な構造となります。その計画・設計は,従来この複雑な相互関係を完全に掌握するため設計図を中心とした2次元的な検討に加えて，プラスチックモデルによる3次元的なエンジニアリングが採用され，信頼性向上がはかられてきました。しかし,こうした手法ではプラント計画完了後のコンピュータを活用した製作設計段階への移行時に，プラスチックモデルから膨大なインプットデータを入力しなければなりません。さらに，最適配置の評価についても,充分な余裕をもてないのが実情でした。

そうした背景の中で，日立はエンジニアリングのスピードアップと最適化の追求をめざして，コンピュータ技術とそのグラフィックモデルによる3次元的表現を縦横に駆使した新しい設計手法を開発しました。それが原子力プラント総合CAEシステム(コンピュータモデルエンジニアリング）です。これは,日立の10数年にわたるコンピュータ・ハードウエアの進歩とともに開発してきた生産設計CADと,今回新たに開発したレイアウト計画CADをエンジニアリングデータベースで結合した画期的な「総合設計・製造一貫システム」です。この導入によって,計画設計のスピードアップから多岐にわたるケーススタディが可能となり最適設計ができるほかさまざまの質的効果が得られています。

今回,「プラント計画及び建設設計へのコンピュータグラフィクスの応用」と題して,配置CADシステムから3次元据付シミュレーションまで，応範囲に展開する総合CADシステムを

(1)基本データ
(2)ルートデータ
(3)レンダリング
(4)シミュレーション
(5)コンポC/G
(6)建屋外工事シミュレーション
(7)建屋内工事シミュレーション

の7回に分けて紹介します。

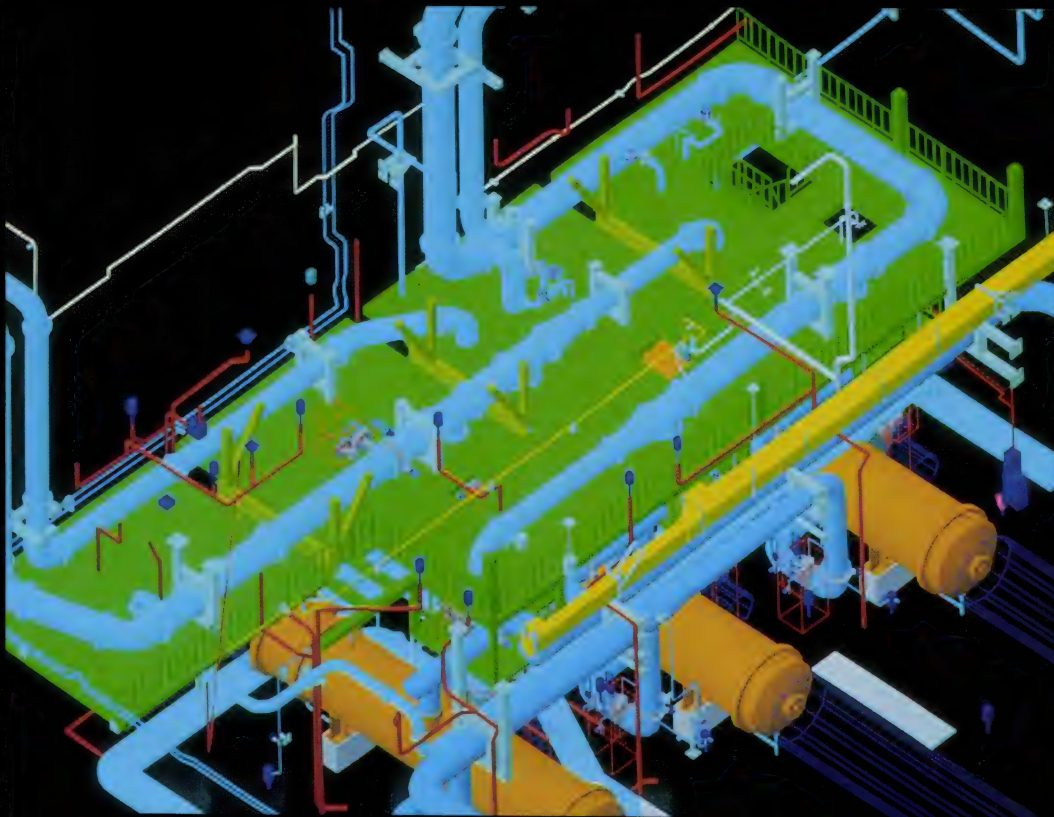



図1　レイアウト計画例

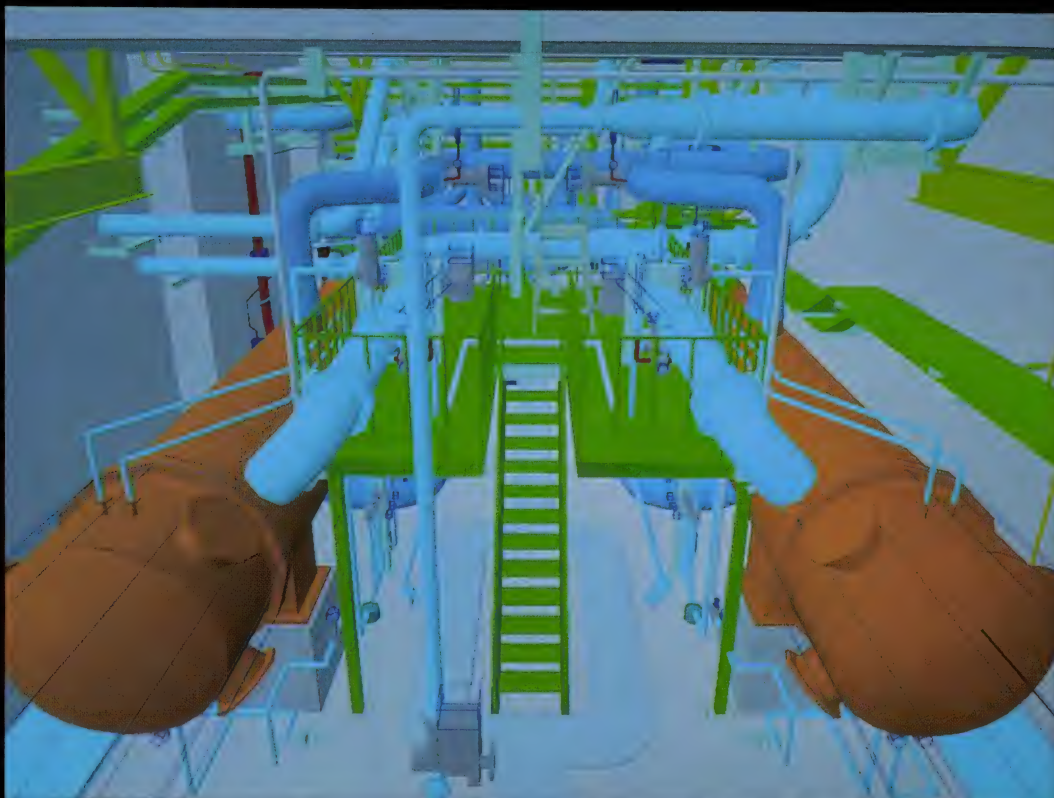



図2　レイアウト計画例

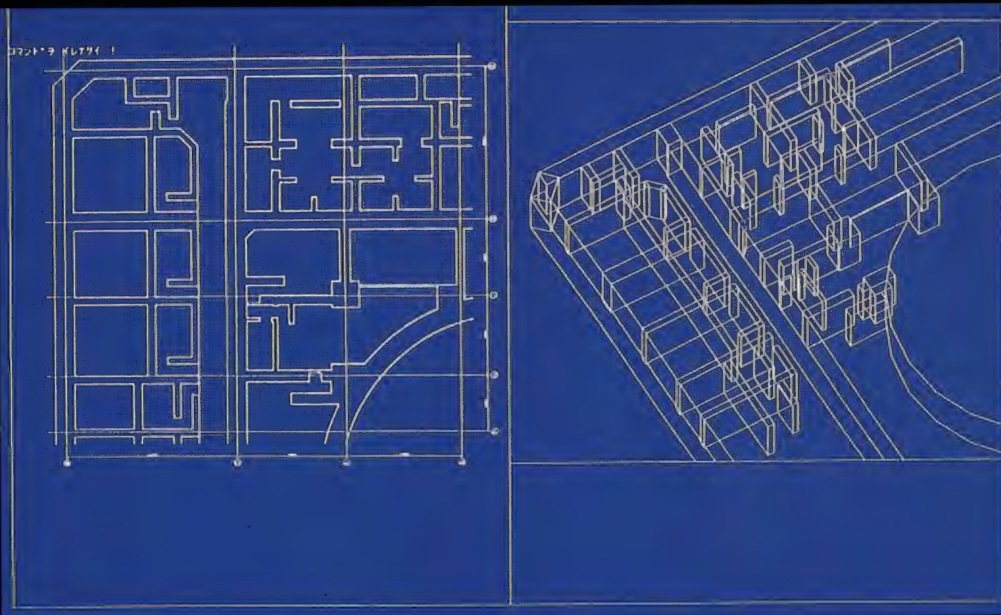



図3　躯体入力例（左は2次元データ，右は3次元データ）

第1回　基本データ

1)躯体入力

躯体入力は2次元躯体図より直接マウスを用いて2次元データを作成して，それに高さを与えることにより3次元の躯体データを作成します。また，躯体以外のもの（例えば，通路スペースなど）に関しても，属性データを入力し分離表示することができます。

プラント計画及び建設設計へのコンピュータグラフィックスの応用

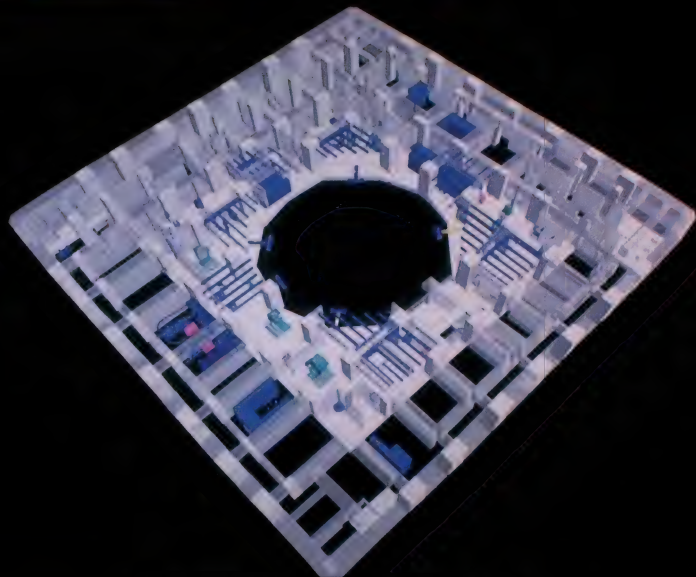



図4　躯体表示例（上記シェーディング結果）

図5　標準機器パターン図

2)機器入力

機器入力は，標準機器パターンより呼び出し，パラメトリック処理により機器形状データが容易に入力でき，また標準機器以外にも任意に形状作成が行えます。

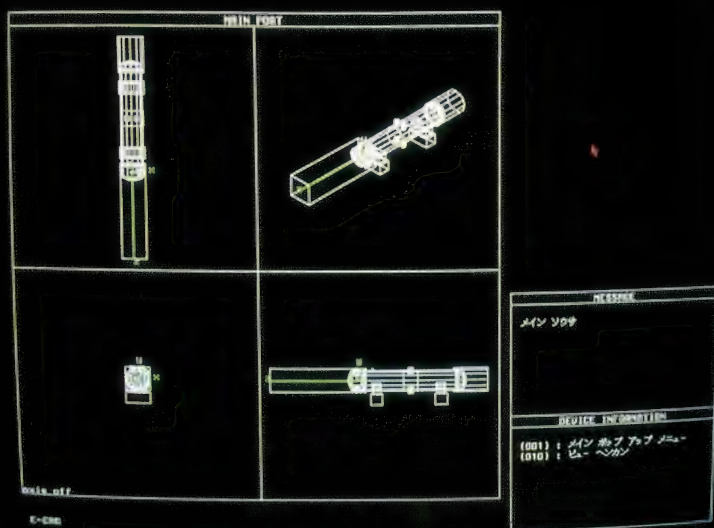



図6　機器入力例（熱交換器）

図7　機器表示例（熱交換器シューディング結果）

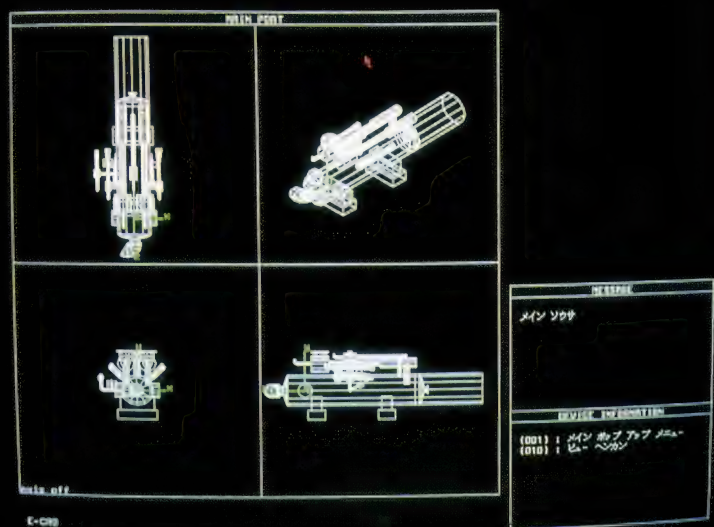



図8　機器入力例（空気抽出器）

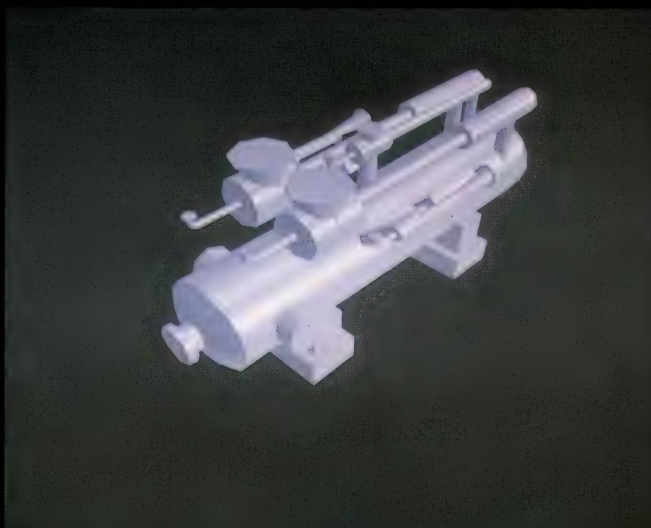



図9　機器表示例（空気抽出器シェディング結果）

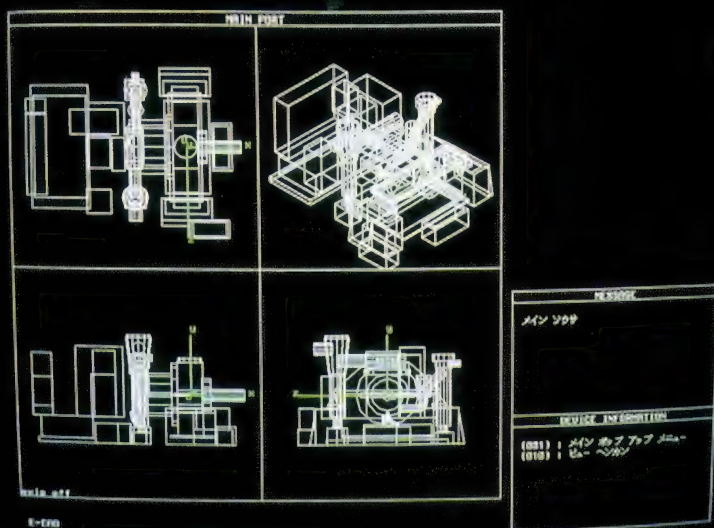



図10　機器入力例（給水ポンプ駆動タービン）

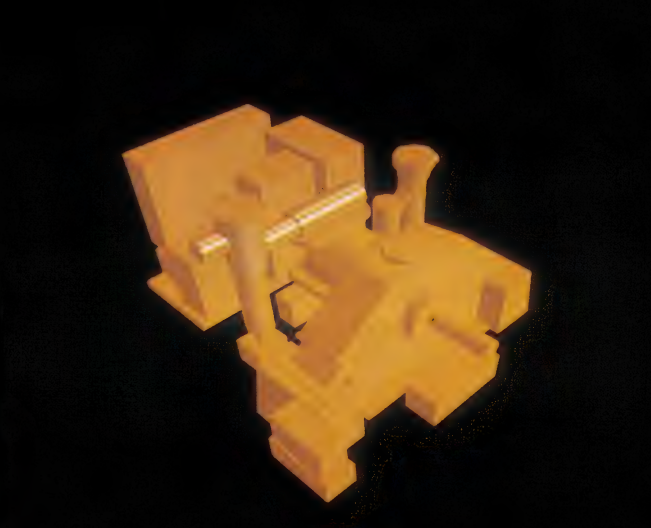



図11　機器表示例（給水ポンプ駆動タービンシェーディング結果）

実用化に向けて動き始めた デスクトップ・パブリッシング

デスクトップ・パブリッシング(DTP),電子出版関連の製品の基本的な枠組みがようやく整い始めてきた。日本語ページ・プリンタの登場,DTPシステムと電算写植機とのインタフェース,カラー・プリプレス・システムの充実といった具合に,ワープロを中心とした閉じた世界から印刷のプロの世界まで踏み込んでいける環境ができつつある。

DTPの中では昨年発表された日本語PostScriptプリンタが出荷を開始し,対応ソフトも増えつつある。日本電気のPC-PR602PS,アップル コンピュータ ジャパンのLaser Writer NTX-JはともにPostScriptに対応し,モリサワの日本語フォントをハードディスクに搭載している。これに対応するソフトウエアとしてMacintosh用ではPage Maker-JとQuark XPressが出されており,PC-98シリーズ用では東洋産業のLinernote PS,Z'ˢ word JG Ver.2.0などが発表されており,今後もさまざまな対応製品が登場してくるものとみられる。

グラフィック・ソフトの中でもAdobe Illustrator88やAldus FreeHandのようにピクセル単位でデータをもつのではなくPostScriptに対応して拡大や変形を自由に行え,4色分解しフィルム出力できるようなものも登場し,デザイナーを中心に大きなブームになっている。従来のピクセル・ベースのペイント・システムとは異なり,高解像度のプリンタで出力すれば高い品質が得られるため,印刷に直結した高品質性と使いやすさが評価されている。

一方,PostScript以外のページ記述言語(PDL)に対応したページ・プリンタ,ソフトウエアもある。代表的なところではすでにかなりの実績をもつキヤノンのレーザーショット用LIPSと,プリンタ大手のエプソンが新たに提案したESC/Pageがある。また,日本電気,アップル コンピュータ ジャパン以外のPostScriptライセンシーの動きも気になるところである。

写植出力を指向した業務用印刷システムおよびソフトウエアも,従来の高価なものから,DTP専用機にソフトウエアで対応したもの,PC-98シリーズなどのパーソナルコンピュータ上で組版をして写研などへデータ変換できるものなどが多数出てきている。

DTP専用機は専用機であるがゆえの高いコスト・パフォーマンスが身上だが,高品位出力のための電算写植機との接続や,入力端末にパーソナルコンピュータやワープロとのインタフェースを考えるようになっており,ワークステーション・ベースの汎用システムに迫るようなネットワーキングができることをうたっているものもある。ホスト・コンピュータとしての機能と今後の能力アップを考えると最も有力とみられるワークステーション・ベースのDTPシステムも,ソニーなどのハードウエア・ベンダーが非常に力を入れていることもあり一層の伸びが期待される。

見逃せないのが,カラー・プリプレスの動きである。米国ではすでに写真のレタッチなどを中心に低価格なプリプレス・システムが登場しているが,日本でもそろそろ各社から製品がそろい始めた。新規需要が急速に伸びているわけではないが,32ビット・パソコンあたりをベースにしたシステムでコスト・パフォーマンスの高いものも出てくるだろう。

以下では,最近の関連製品の中から特徴のある製品を中心に取り上げてみた。

SYS

UA-100 EX

㈱いづみや

総合文書処理システムとうたっており,DTPシステム機能ももっているが,作成したデータは写研の電算写植システムへ出力するためのファイルにも変換できる。各種ワードプロセッサで作成された文書をMS-DOSテキスト形成で取り込んでレイアウトを行う。出力は専用のページプリンタへも可能。価格は398万円。

問合せ先

〒150 東京都渋谷区恵比寿1-20-8
☎03(440)1511

Compotex 3000, Compart

印刷機械貿易㈱

ともに印刷用の組版システム。Compotex 3000は1,200 dpi

で印画紙への出力ができる。ホスト•コンピュータにNEWSを採用したためネットワーク化も可能。Compartは，PC-98 XL²ベースのワークステーションと600 dpiの解像度のレーザービーム・プリンタから構成されている。両システムとも直接，実際の業務で版下や電算写植のデータとして出力することができる。フォントは，明朝体，ゴシック体，欧文10書体を独自のアウトライン•フォントでもつ。DTPというよりは，むしろ印刷業界などのプロのオペレータが使うことを目的に設計されたものである。価格は，Compotex 3000が標準で約5,000万円，Compartが標準で約900万円(電算写植で出力するにはオプションでプラス約1,600万円)。

問合せ先

〠140　東京都品川区南大井3-21-4

☎03(763)4141

NICVECTOR 7/CD, NICVECTOR 21/CD

エヌアイシー販売㈱

ホスト・コンピュータにはYHPのvectraを使用している。これに専用LSIボードRICHを組み込んだDTPシステム。プリンタはゼロックスのレーザービーム・プリンタを採用。独自のアウトライン・フォントによりWYSIWYG編集ができる。画面上での編集はカラー対応であるが，出力装置はモノクロしか対応していない。電算写植へのデータ変換はできないが，ユーザーからのニーズがあれば作成する。NICVECTOR 21/CDは写真から入力を行って編集できる。価格はNICVECTOR 7/CDが780万円，NICVECTOR 21/CDが3,075万円。

問合せ先

〠160　東京都新宿区西新宿8-11-3

☎03(348)1831

PRIME Composition Table

大倉電機サービス㈱

ホスト•コンピュータには東芝AS 3000を採用し，印刷業種，新聞社を対象とした電子組版編集システム。入力は1,016 dpiのスキャナを用いる。ソフトウエアは，文書編集，版下作図，定形組版，カット見出し部から構成されている。出力は，最高600 dpiのレーザービーム•プリンタに版下イメージを，または専用の半導体レーザーの電算写植に対してのいずれも可能。価格は約1,000万円。

問合せ先

〠104　東京都中央区銀座2-3-6

☎03(274)1943

スタジオ9500

㈱大阪ヤマトヤ商会

英国クロスフィールド社のカラー集版ワークステーション。大容量の記憶装置に大量のイメージデータを格納できる。価格は未定。

問合せ先

〠530　大阪市北区南森町2-3-9

☎06(313)3352

J-PERQシステム

㈱学習研究社

画像レイアウト・システム。各種ワードプロセッサ，パーソナルコンピュータからのデータ入力が可能，各社の出力機へ出力ができる，WYSIWYG編集などの特徴がある。データは文字，図形，画像などを同じように扱えるマルチメディア対応。電算写植システムへの出力も可能である。価格は基本システムで約1,000万円。

問合せ先

〠146　東京都大田区仲池上1-17-15

☎03(726)8511

EZPS 3200, キヤノワードPRO 1000

キヤノン販売㈱

両機種とも同社独自のDTPシステムであり，ワードプロセッサと連結されている。EZPSはすでに実績があるDTPシステム。文書データ・フォーマットは独自のものを使用する。出力は同社のレーザーショットではなく専用のレーザービーム・プリンタ（400 dpi）を採用。アウトライン・フォントをホスト•コンピュータの40 Mバイト•ハードディスクにもっており，WYSIWYG方式による編集が可能。価格は，EZPS 3200が38万円，PRO 1000が198万円。なお，電算写植へのデータ変換ソフトはEZPS 3200用のみ東洋産業から販売されている。

問合せ先

〠108　東京都港区三田3-11-28

☎03(455)9060

FX 550 SA, FX 550 LAN システム

㈱クスダ

両製品とも印刷向けの DTP システム。LAN システムはネットワーク対応ができ，SA はローコストが特徴。19 インチ CRT で WYSIWYG 編集ができる。文字サイズは 2 ～32 mm 間の指定が可能。出力は 480 dpi のレーザー・プリンタまたは印画紙，フィルムへ行う。価格は，ワークステーション，入出力サーバーおよびプリンタから構成される FX 550 の基本システムが 550 万円。各システムはこれを基本に構築していく。

問合せ先
〠105　東京都港区西新橋 1 - 5 -11
☎ 03(502)3514

レーザーコンプ，デジタルビジョン

㈱小池製作所

写真と文字を同時にページアップできる電算出力機。出力はヘリウムネオン・レーザー方式を採用，1 インチ当たり 1,000 本の高密度な出力ができる。文字書体専用に 300 M バイトのハードディスクを 2 台接続できる（1 台で欧文 1,024 書体)。価格は，レーザーコンプが基本構成で 2,600 万円，デジタルビジョンが 512 万円。

SAIVERT-H202

問合せ先
〠146　東京都大田区鵜の木 3 -23-18
☎ 03(758)2361

Performik

コニカ㈱

32 ビット・コンピュータを複数接続するマルチ処理により生産性を高めることができるトータル製版システム。マルチシステムは，レイアウト・文字組版ユニット，画像処理ユニット，集版ユニット，入力ユニット，出力ユニットの 5 ユニットが，それぞれ 32 ビット CPU をもち独立して動作する。ファイルは共通のファイル・ユニットによって共有化されている。また，作業に合わせて各ユニットのうち必要な部分のみを増設できる。価格は標準システムで 1 億 8,800 万円となっている。

問合せ先
〠163　東京都新宿区西新宿 1 -26- 2
☎ 03(349)5155

プログラムフリープロセッサ

三洋電機㈱

OCR（光学式文字認識装置）入力システム，OLC 1（手書き文字および図形認識キーボード)，300 dpi のレーザー・プリンタで構成されるシステム。データベース，表計算，ワードプロセッサなどの機能が組み込まれている。価格は未定。

問合せ先
〠550　大阪市西区江戸堀 2 - 7 -25
☎ 06(443)5144

SAIVERT-H 202, HITCAP-SC

㈱写研

SAIVERT - H 202 は写研の一連の電算写植システムで，WYSIWYG 編集が可能。システムは，レイアウト・ステーションの SAIVERT-H 202，入力校正機 SZANNA-SP 313，組処理・多書体内蔵の普通紙プリンタ SAGOMES-GL 621，画像入力装置 SAPGRAPH-171，イメージタイプ・セッター SAPLS-LauraSS などから構成されており，別々に購入またはレンタルすることも可能。すでに多数のユーザーで実際の業務に使用されている。本体のみの価格は 70 万円で，業務内容により異なる。

HITCAP-5Cは写研データへ出力ができるDTPシステムで，日立のDTPシステムHITCAP 500をベースにシステム化されたものである。画面上では明朝体とゴシック体の2書体のみしか表示しないが，写研書体を指定することによって1つの文書の中で最大24書体まで使い分けることができる。編集が終わったらすぐ写植データとして出力できる。文字のサイズは写植と同じ級数単位であるため，写植機と同じオペレーションで操作できる。写研のDTPシステムは本機が初めてである。価格は未定。

問合せ先

〒170　東京都豊島区南大塚2-26-13

☎03(942)2211

マッチプリント

住友スリーエム㈱

米国3M社のマッチプリント(色校正)システム。すでに米国およびヨーロッパで実績がある。現在，日本で一般的に使用されている色校正方式の平台校正機と異なりポジ色校正方式を採用しているため，安定した仕上りが得られる。価格は1セット395万円。

問合せ先

〒158　東京都世田谷区玉川台2-33-1

☎03(709)8274

DIC-QP 2000

大日本インキ化学工業㈱

カラー画像レイアウトと文字組版が同時に同一画面で行えるカラーページ・デザインシステム。ラフスケッチの段階から最終印刷物と同じクオリティのレイアウトが得られる。データは光ディスクに収められ，レイアウト修正も容易。また，自動版下製図カッティング・システムLAYALLも販売している。価格は，QP 2000が約2,300万円，LAYALLが約1,800万円。

問合せ先

〒101　東京都千代田区外神田2-16-2

☎03(258)3660

電字9032-Σシリーズ

電字システム㈱

ホスト・コンピュータに東芝AS 3000を採用した電算写植システム。WYSIWYG編集ができるソフトウエア，600 dpiレーザー・プリンタのページアン，1,016 dpiイメージスキャナのピリカ，1,016 dpiレーザー・プロッタのイメーザなどの高解像度の周辺機器を使用。出力する写植文字は5,000 dpiに相当する高解像度であり，これらのデータを写植出力コントローラを使って直接印画紙に自動出力できる。価格は手動写植機の標準構成システムで960万円。

問合せ先

〒101　東京都千代田区岩本町2-14-19

☎03(862)5211

シグマグラフ システム6000，IP-700

大日本スクリーン製造㈱

シグマグラフシステム6000は1つのテーブルで図形作成・切抜き・レタッチ・貼込み・検版までを行うことができる多目的集版機。IP-700はWYSIWYGにより完全版下の作成ができる。フォントは独自のアウトライン・フォントを採用，図面データと同じようにして扱える。価格はIP-700が8,500万円。

問合せ先

〒170　東京都豊島区東池袋5-38-7

☎03(989)3631

AS-Documents

㈱東芝

ホスト・コンピュータに同社EWSのAS 3000を採用したDTPシステム。文書データ・フォーマットは独自のもの。出力は，A3サイズ400 dpiまたはA4サイズのページプリンタに対して行える。AS 3000にアウトライン・フォントをもっているためWYSIWYG方式の編集が可能。システムはホスト・コンピュータ，プリンタ，ソフトウエア(OS含む)で構成されている。価格はクラスによって約500万～2,100万円程度。

問合せ先

〒105　東京都港区芝浦1-1-1

☎03(457)2725

レスポンスシステム

日本サイテックス㈱

デザイン，製版の業務を一貫してコンピュータ化できるシス

テム。デザイン段階でのチェックやデザインのバリエーションにも対応できる。価格は，標準システムで4,500万円程度から。

問合せ先

〒104　東京都中央区京橋3-14-6

☎03(563)0761

PROST

日本情報サービス㈱

教育専門出版社である新学社で実際に使われている編集支援システムを一般向けに商品化したもの。編集者など組版専門のオペレータでなくとも簡単に操作できることを目的として開発された。オペレーションはレイアウト方式を中心にファンクション方式を採用したため，原稿を異なるレイアウトに流用してもほとんど修正の必要がない。ホスト・コンピュータはアポロのDOMAINを採用し，高速な組版処理が可能。標準システム価格は700万〜800万円。

問合せ先

〒550　大阪市西区新町1-6-3

☎06(536)7621

文作30 DTP

㈱日本デジタル研究所

従来の文作シリーズの印刷・編集機能を強化したもの。ホスト・コンピュータはMC 68020とMC 68881を採用した32ビット版。WYSIWYG方式の編集が可能である。アウトライン・フォントは独自開発のものを採用。出力データは独自のPDLフォーマットをレーザーショット（400 dpi）に出力する。オプションで400 dpiの解像度のイメージリーダ（スキャナ）が接続できる。価格は225万円。なおデータ・フォーマットは同一だがページ編集機能がない文作30（179万円）もある。

問合せ先

〒163　東京都新宿区西新宿2-1-1

☎03(348)6751

N 5170 モデル55

日本電気オフィスシステム㈱

トータル印刷システム。クラスタシステムによる分散処理を可能として，組版処理の高速化を実現。日本電気のPostScript対応のDTPの流れとは別のDTPシステム。編集はWYSIWYG画面で行うことができる。これまで，大きな文書を扱うときは複数のフロッピに分け，実際の原稿は出力した後に切貼り処理を行っていたが，マルチ原稿プリント機能ができたためにデータは自動的に1文書化されて出力される。出力は480 dpiの普通紙プリンタの他，印画紙またはフィルムに723 dpiまたは1,446 dpiの解像度で出力できる。価格は最小構成で940万円。

問合せ先

〒108　東京都港区芝4-13-2

☎03(769)8605

HITCAP 500

㈱日立製作所

企業内文書などの作成のためのDTPシステム。これまで高品質名文書は外部の専門業者への依頼などでコストがかかっていた会社などに最適。データはイメージスキャナから読み込んだ写真などのイメージデータも同時に扱うことができる。フォントはベクトル・フォントも準備されている。文字サイズで4〜150級までの大きさを15.7ドット/mmのプリンタに出力できる。標準的なシステム価格は398万円。

問合せ先

〒143　東京都大田区大森北1-18-2

☎03(763)2411

EG-pub.

富士電機総設㈱

入力したデータを瞬時に画面表示するWYSIWYG機能を備え，レイアウトも迅速に行える。400 dpiのレーザー・プリンタへ出力する。フォントは明朝体，ゴシック体のベクトルと欧文5書体を備えている。価格は，ホスト・コンピュータがFMR-60またはFMR-70，レーザー・プリンタのサイズなどで314万3,000〜443万1,000円。

〒104　東京都中央区八丁堀1-9-6

☎03(552)7641

IPS/GX

富士通㈱

印刷，出版，新聞などにおける文書編集から，組版・出力までをマルチメディア環境で行うことができるシステム。Mシリーズをホスト・コンピュータとして，2〜4台の編集用ワーク

ステーションをオンライン接続することにより，文書やデータの一括管理ができる。文字・図形・線画・写真を同時に扱うことができ，新聞などの版下を切貼り処理を行わずに作成できる。専用の 960 dpi の解像度の IPS レーザー・セッターでフィルムや印画紙に印刷できる。価格は 2 ワークステーション・タイプで 1,700 万円。

問合せ先

〒100　東京都千代田区丸の内 1 - 6 - 1

☎ 03(216)3211

Panapage

松下電器産業㈱

社内での印刷スピードアップや経費節約などに有効な DTP システム。元来印刷業界向け電子組版システムであったが，アマチュアにも使用できるような簡単操作を実現した。出力は，400 dpi のレーザー・プリンタへ行う。また，同社の印刷業界向け電子組版システムとデータの互換性があり，1,016 dpi で版下の出力もできる。価格は標準セットで 300 万円。

問合せ先

〒105　東京都港区芝大門 1 - 1 -30

☎ 03(535)6261

QuBix，LASER 7 EX

㈱モトヤ

QuBix は印刷分野向けの文字・図形処理システム。LASER 7 EX は入力・校正・編集の一連の処理を行う電算植字システム。価格は，QuBix が 1,670 万円から，LASER 7 EX が 300 万円から。

問合せ先

〒542　大阪市中央区船場 1 -10-25

☎ 03(959)0123

レーザイメージセッター IM 30

㈱モリサワ

写研に次いで 2 番目のシェアをもつ電算写植メーカーであるモリサワの IM 30 は，同社のデジタル・フォントのノウハウを生かした新しい形式のフォントによる RIP を採用した。フィルム・プリンタおよび普通紙プリンタに出力できる。フィルム・プリンタへは 1,400 dpi の解像度で，普通紙プリンタへは 480 dpi の解像度で出力できる。標準的な価格は 1,200 万円から。

問合せ先

〒162　東京都新宿区下宮比町 15- 5

☎ 03(267)1231

RECS 200，REONET 300

リョービイマジクス㈱

RECS 200 は従来の手動写植機の文字盤がそのまま使えるアナログ出力の電子組版システム。組版校正機，出力機などから構成されている。REONET 300 は文字・画像をページアップ状態で同時出力できる電算写植機。価格は，標準システムで，RECS 200 が 1,600 万円，REONET 300 が 1,700 万円。

問合せ先

〒101　東京都千代田区外神田 3 -15- 1

☎ 03(257)1241

ソフト

WARX-1，WARX-2

㈱アルクス

PC-9801 で動作するデータ変換ソフトウエア。MS-DOS のテキストデータと写研データ（SK 72，SK 78 コード）のデータ変換を行う。WARX -1 はテキストデータのみであるが，WARX- 2 は写研文字をすべてもっている。変換は双方向で可能。価格はセットで 64 万 5,000 円。

問合せ先

〒170　東京都豊島区池袋本町 2 - 8 -22

☎ 03(981)3250

EGBook Ver 1.3

㈱エルゴソフト

現在，国産唯一の Macintosh 用ページレイアウト・ソフト。LaserWriter に対応。独自のアウトライン・フォントをもっているが，LaserWriter のフォントを使用することもできる。独自のフォントを使用した場合，縦書きなどのレイアウトを行った際に美しく出力できる。LaserWriter のフォントを使用する場合は漢字トーク 6.0 が必要。データは，現在は PICT（Mac のプリンタ出力のデータ・フォーマット）形式のみ出力でき，PostScript の出力はテキスト文字のみとなっている。電算写植への

データ変換はできない。価格は6万8,000円であるが，テキストの入力はワードプロセッサの方が効率が良いためMac用ワープロ・ソフトEGWordとセットで9万8,000円の購入を勧めている。

問合せ先
〒105 東京都港区元赤坂1-2-5
☎03(589)4951

暁，スーザン98，助さん，他

秀和システムトレーディング㈱

暁は，PC-98のMS-DOS上で作成･編集された一太郎やMS-DOSの文書を，写研のSKコードに変換するソフトウエア。スーザン98は写研のコード機能を入力，校正することができるソフトウエア。助さんはOASYSと一太郎の文書コンバータ。同社では，電子組版システム表組くん98，電子組版支援システム版下工房98などを販売している。価格は，暁が220万円，スーザン98が21万8,000円，助さんが1万4,800円。

問合せ先
〒107 東京都港区赤坂8-5-29
☎03(470)4941

Z'sword JG Ver.2.0

㈱ツァイト

PC-98用ページレイアウト・ソフト。出力プリンタが日本電気のPC-PR 602 PSの場合は，プリンタ側のアウトライン・フォントを使用。出力コードも日本語PostScriptとなっている。ただし，ベジエ曲線オペレーションには対応してない。604 PS以外のプリンタ（PC-PR 201など）の場合は，ソフトウエアが独自にもっているアウトライン・フォントを使用してイメージ展開し出力する。ソフトウエアのアウトライン・フォントを使用する場合4Mバイト以上のRAMディスクが必要。編集画面では文字はPCの16ドット文字を使用するが，行間などのページ・レイアウトはできるだけ出力イメージを忠実に再現する。予定価格は5万8,000円。

問合せ先
〒150 東京都渋谷区初台1-47-1
☎03(299)0463

Quark XPress

システムソフト㈱

米国Quark社のMacintosh用のページレイアウト・ソフト。LaserWriterに完全に対応する。入力データはPICT以外に，EPSF（Encapsulated PostScript File：Macにおける図形のデータ・フォーマット），TIFF（Tag Image File Format：Macでスキャナから読み込む際のデータ・フォーマット）などに対応している。EPSF，TIFFともApple，Aldus，Adobeの各会社で標準として決定されたものである。縦組のレイアウトができる他，文字を含んだカラー表示にも対応している。またワープロ機能，オート・ハイフネーション機能など現在入手できるMac用日本語DTPソフトウエアの中では最も完成度が高い。動作環境は漢字トーク2.0以上。価格は18万円。

問合せ先
〒800 福岡市中央区天神5-7-2
☎092(752)5269

SMI/EDIAN

住友金属工業㈱

ホスト･コンピュータにはNEWS，ソフトウエアは管理工学研究所開発のDTPシステム。商業印刷にも使用できるが，一般の企業，官庁，学校，研究機関などでの業務用文書の印刷をターゲットとしている。NEWSのもつネットワーク機能とファイル管理機能を生かして各種データの共有，一元管理が行える。操作はマウスとアイコンで行うことができる。フォントはNEWSの拡張アウトライン･フォントを使用。プリンタ，スキャナなどの周辺機器はNEWSのものをそのまま使用できる。電算写植への出力が可能。価格はソフトウエアのみで300万円。

問合せ先
〒100 東京都千代田区大手町1-1-3
☎03(282)6715

SMI/EDIAN

サザンナ

㈱トータルメディア研究所

各社のワードプロセッサから共通方式で写研の文字やファンクションが入力できるソフトウエア。従来，機種別に分かれていた写研の文字およびファンクションの入力方法が一元化できる。価格は9万8,000円。写研外字セットは1万円。

問合せ先

〒173　東京都板橋区向原1-11-6

☎03(554)0636

Linernote PS，EZFIT

東洋産業㈱

PC-98用DTPシステム。日本語PostScriptコードの出力ができ，PC-PR 602 PSのPostScriptに完全対応。ソフトウエアは縦組も含めたページレイアウトができ，1ページ当たり1万6,000ベクトルまで管理できる。また，1ページ当たり8レイヤーの用紙を重ねることもできる。ホスト・コンピュータはPC-9801 RAであるが，専用の画像処理プロセッサを使っているため高速な画像データの処理が可能。価格は専用の処理ボードとソフトウエアで110万円。同社ではPostScript対応ではなく，専用のアウトライン・フォントを使い独自のデータ形式のLinernoteも販売している。EZFITはキヤノンのDTPシステムEZPS 3200のデータを写研のSK 72データに変換するソフトウエア。価格は118万円。

問合せ先

Linernote PSの出力例

〒101　東京都千代田区神田須田町2-25

☎03(256)7355

NICEシステム

㈱日本情報処理開発

シャープのワードプロセッサ書院を入力端末として使う電算写植システム。電算写植システムへのデータ変換を行う電植ソフトウエアNICE SHSK，電算写植データのレイアウト確認を行うイメージャNICE EPONAなどから構成されている。価格は標準的なシステム構成で680万円。

問合せ先

〒105　東京都港区虎ノ門1-11-2

☎03(591)3407

Publiss

日本電子計算㈱

ホスト・コンピュータにNEWSを採用したDTPソフトウエア。データ形式はSCR形式（SCR社（北海道）のデータ・フォーマット）であるが，近い将来PostScriptへの対応も行う予定である。出力は，NWPシリーズ（400 dpi）またはレーザー・ショットシリーズ（240 dpi）のいずれかを構入時に選択。将来PostScript対応がリリースされた場合はPostScriptが選択できるようになる（1ソフトで複数プリンタ対応は不可）。漢字は独自のアウトライン・フォントをもっており，画面上ではA4サイズ用紙のイメージのまま編集ができる。オプションで一太郎やOASYSのデータを取り込むこともできる。価格は60万円。基本的にX-Window上で動くがウィンドがなくても可。

問合せ先

〒103　東京都中央区日本橋兜町6-7

☎03(241)5791

秀嶺LEVEL III

日本パントール㈱

PC-9800シリーズで動く組版ソフトウエア。最終的には写研へ出力を行う。松，一太郎などと連動して動かすことができる。価格は220万円。

問合せ先

〒101　東京都千代田区神田神保町2-32

☎03(237)7003

Page Maker-J

㈱ブリッジ

米国アルダス社開発の Macintosh 用のページレイアウト・ソフトウエアの日本語バージョン。DTP という言葉自体を生み出したソフトウエアであり，日本語 PostScript に完全に対応している。ワープロ入力ができ，英語のハイフニング機能をもっている。データは，アスキー形式，PICT，TIFF，EPSF などの Mac 標準データの他，制限付きで EGword（エルゴソフト）のデータにも対応している。価格は 12 万 5,000 円。

問合せ先

〠101　東京都千代田区神田北乗物町 11

☎03(258)7357

DTP 4 U，もじら，PMS 88

㈱ロジスティックス

すべてデスクトップ・プレゼンテーション・ソフトウエア。SCR のアウトライン・フォントを使って文書（イメージ）を作成する。DTP 4 U はフルカラーでビデオへ，もじらはレーザー・ショットへ，PMS 88 はスライドなどへ出力する。すべて PC-98 で動作する（PMS 88 はハイレゾモードのみ）。価格は，DTP 4 U が 34 万 8,000 円，もじらが 9 万 8,000 円。

問合せ先

〠162　東京都新宿区神楽坂 2 -13

☎03(718)8799

ハード

LaserWriter II NTX-J

アップル コンピュータ ジャパン㈱

CPU に MC 68020 を採用した日本語 PostScript 対応のページプリンタ。英語アウトライン・フォントは ROM で，日本語アウトライン・フォントは内蔵ハードディスクでもっている。古い LaserWriter シリーズからのバージョンアップはボードを取り替えるのみでできる。1 台のプリンタを 31 台の Mac で共用可能。解像度は 300 dpi。価格は 119 万 8,000 円。

問合せ先

〠106　東京都港区六本木 1 - 4 -30

☎03(224)7002

デジタルカラープリンタ

大日本印刷㈱

実際の印刷と同じレベルのフルカラー印刷ができるハードコピー。昇華型熱転写方式を採用。また多数の独自の 3 次元文字フォントを内蔵しており，デザインにおけるバリエーションに対応できる。CG 画像をそのまま印刷する CGPS も準備されている。価格は 660 万円。

問合せ先

〠162　東京都新宿区市谷加賀町 1 - 1 - 1

☎03(266)2655

アートビット 101

㈱東京グラフィック・アーツ

CCD ラインセンサー型カメラを使った図形入力ターミナル。PC-9800 シリーズなどのパーソナルコンピュータに接続して使用する。従来のデジタイザ方式に代わって写真原稿などを CCD で入力できる。価格は 500 万円から。

問合せ先

〠171　東京都豊島区高田 3 - 5 -13

☎03(984)3591

CALIBRATOR, TPG-4300

㈱東陽テクニカ

CALIBRATOR は高解像度のカラーモニターで，価格は 140 万円。TPG-4300 はフルカラー・ビデオプリンタで，価格は 198 万円。

問合せ先

〠103　東京都中央区日本橋本石町 1 - 1 - 2

☎03(245)351

FX 550

東レエンジニアリング㈱

完全版下を作成できる電子組版システム。システムはワークステーション（編集・校正機）と入出力サーバー，レーザービーム・プリンタから構成されており，LAN 機能によりワークス

FX550

テーションが最大8台，ハードウエアの増設とネットワーク化が図れる。入力データは一太郎，OASYS，JWなど各種ワープロから可能であり，出力は専用の480 dpiのレーザービーム・プリンタの他，写研の電算写植システムへのものも開発中。価格は380万円。

問合せ先

〒103　東京都中央区日本橋本石町3-3-16

☎03(241)8461

Focus S 600/800, CG 9400 PS

日本アグファ・ゲバルト㈱

すべて米国コンピュグラフィック社の製品。Focus S 600／800は解像度600／800 dpiのDTP用画像入力スキャナ。Macintoshなどの SCSIインタフェースをもつコンピュータに接続できる。CG 9400 PSはPostScript対応のレーザーイメージ・セッター。市場の多くのPostScript対応のフロントエンド・システムと接続が可能。出力解像度は1,200／2,400 dpiの切替え。価格は，S 600が100万円，S 800が120万円，CG 9400が980万円。

問合せ先

〒105　東京都港区虎ノ門3-8-21

☎03(438)1777

VT 600

日本エーエム㈱

PostScriptレーザーイメージ・セッター。600 dpiの解像度をもつ。Macintosh，IBM　PCなどの多数のPostScript対応のDTPシステムに接続可能である。英文フォントの35書体が標準インストールされているが，オプションで4書体・3万円程度で増設できる。日本語のアウトライン・フォントの対応は未定。価格は375万円。

問合せ先

〒102　東京都千代田区麹町5-3-1

☎03(263)0841

PC-PR 602 PS

日本電気㈱

CPUにMC 68020を採用した，日本語PostScript対応のページプリンタ。日本語アウトライン・フォントは内蔵の40 Mバイトのハードディスクに格納されている。解像度は320 dpi。また従来のPC-PR系のエミュレーション・モードをもっている。価格は98万円。

問合せ先

〒108　東京都港区芝5-33-1

☎03(454)5111

フォント

PostScript対応フォント

㈱アイコン

2(6ポイント)～300 mm四方の文字が出力できるアウトライン・フォント。文字種は現在，明朝体，ゴシック体，江戸文字が販売されているが今後増やしていく予定で，最終的には12種類，60書体が提供される。価格は，字母として切り売りする場合で200万～5,000万円。ロイヤリティ販売も行う。

問合せ先

〒185　東京都国分寺市本町3-1-22

☎0423(26)0085

漢字シンセサイザー

あとりえフォンタップ

パーソナルコンピュータ(PC-98 XL²など)やEWSでアウトライン・フォントを作成するシステム。デザイナーが基本的な字画をデザインすれば，後はオペレータがフォントを作成していくことができる。文字使用関連業種への販売を行う。価格は，パーソナルコンピュータ用が510万円，EWS用が980万円。

問合せ先

〒170　東京都豊島区巣鴨1-3-1

☎03(946)3677

ゲームの中のCG

わずか十数年で，メカニカルゲームを駆逐したビデオゲームはますますその面白さを増している。それを支えているのはもっぱら電子技術の進歩だ。そしてとうとう3次元CGを利用するものまで出現した。限りなく人工環境に接近するビデオゲームの歴史と現在を眺めてみよう。

従来，ビデオゲームの画面はディスプレイ技術の一つの極限を追求した形と考えられ，狭義のコンピュータ・グラフィックス（CG）とは接点がなかった。しかし，ゲームに要求される表現内容は必然的にCGに結び付いていかざるを得ないものであった。最近，この“約束された結合”を実現した製品が相次いで出現した。そこで，コンピュータ・アプリケーションの一つとしてビデオゲームが出現してから最近のCGを使ったゲームまでを振り返るとともに，併せてゲームとCGのかかわりについても改めて考えてみよう。

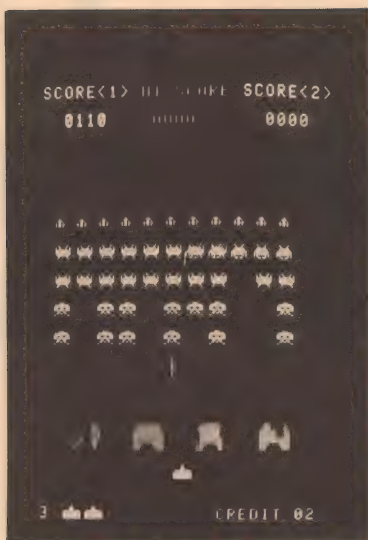
ゲームマシンの歴史

アミューズメント・マシンの歴史は長い。ギリシャ時代の神殿でのかがり火による自動開閉扉，江戸時代のからくり人形，戦後おおいに流行ったのぞきからくりなど，遊興施設とそこに設らえられた設備の関係にみられるように，古今東西を問わず多くの人が集まるところにアミューズメント機能をもった機械類が発達してきた。この流れの中ではっきりとゲームマシンの形態を確立した最初のものは，おそらくアメリカのカジノマシンの中のある種のものだろう。カジノの設備は，①ポーカーのようにもっぱら道具を使って人間同士が対話的に行うもの，②ルーレットなどのようにマシンを使うがそれを操作する人間にポイントを置いているもの，③スロットマシンのようにプレーヤーが機械だけを相手にしてプレーするもの――に分類できるが，ここで述べようとしているのは③である。

最初カジノマシンとして出現したアミューズメント・マシンには，先に述べたスロットマシンやピンボールなどがある。これらは最初，賭博機械として発達したのだが，しばらくするうちに酒場の余興に使われるゲームマシンとして定着した。ここから本格的なゲームマシンの歴史が始まったと考えられる。いったん始まってからは普及・発達がきわめて速かった。大人から子供までその対象を広げ，内容的にもドライブゲームやシューティングゲーム，野球やゴルフなど現実のイベントやスポーツをシミュレートするという現在のゲームの要素は，当時すでにほとんどが出現していた。

しかし，初期（50年ほど前）のゲームマシンはその動作メカニズムを単純な電気機械式（単純なスイッチやリレーで結構複雑なものを実現していた）によらざるを得ず，ゲームのリアリティーを支えるリアクションももっぱらメカニズムによるものであった。そのためシチュエーションやストーリーは大きな制約を受け，ゲームに馴染む対象とそうでないものとのギャップを埋めることはできなかった。

その後ゲームの技術的要素が研究され，その結果としてマサチューセッツ工科大学（MIT）でビデオゲームが開発された。これはTTL（トランジスタ論理回路）で構成されており（CPUはなし），四角い棒や丸形のシンボルを表示しながら記号化された抽象の世界に遊ぶといったもので，それまでのゲームの流れである具象のシミュレーションとは一線を画するものであった。ある意味ではエレクトロニクスの助けによって，初めて具体的現実と離れた抽象世界を対象とした遊びの新しい形式が加わったともいえよう。この種のゲームは最初アタリ社によってアーケードゲームとして発売され，その後ぞくぞくとバリエーションを増やしていった。この中には多くのファンを獲得したテーブルテニスやピンボールゲーム，ブロックくずしなどの白黒ビデオゲームがあった（技術的にカラー表示は難しかった）。日本ではこのころからビデオゲームを生産するようになった。

続いてアメリカで世界最初のCPUを使用したゲームが出現

タイトーの初期のCPUゲーム「スペースインベーダー」のゲーム中画面

した。白黒ではあるがCPUの使用によって多くの条件分岐や例外処理を実現し,対話的に得られる情報量が増えた。これによりゲームマシンの差別化は質的な内容を競うものに変化していく。日本でも早速,類以のものが作られるようになった。その代表的なものにスペースインベーダーがある。これはインテルの8080を使った2番目のCPUビデオゲームで,日本のアーケード・ビデオゲームのあけぼのとなった。最初に白黒版が作られ最終的にはカラー化されたが,ビットマップ・ディスプレイを使っていたので画面の高速な書換えは難しく,そのスピードに見合ったゲーム内容を模索することが結果的にゲームのコンセプトを煮詰めることにもなった。また,このあたりから日本製のゲームマシンが世界へ出荷され始めた。そして高速化を果たしてカラー化されたインベーダーや,新しく出現したパックマンなどが輸出されるようになると,世界のビデオゲームの開発は日本がイニシアチブを取るようになった。

画面の高速な書換えを実現するためハードウエアによるスプライトが出現し,ギャラクシアンなどのより技術的に進歩したきれいな映像をもつゲームが実現した。このときから現在まで約10年の間,基本的にはこの大枠の中での進歩が追求された。また,メインチップはMC 6800からMC 68000へ(アメリカでは最初8080 AやZ 80 A,6502も使われた)世代交代した(ゲーム用には68系のチップが好まれ,現在はMC 68020/MC 68030への交代途上にある)。

ここまでは画面表示に奥行きがなく,俯瞰的な視点が当たり前であった。しかしプレーヤーからはもっと主観的な視点を求める声が高かった。そこでインベーダーから4~5年後には疑似的に水平方向の視線と動きを実現するものが現れた。表示する物体のデータを大幅に増やし,同じキャラクタに対してもその距離によって異なる大きさの形状データを使用するのである。この新しい視点感覚はプレーヤーの共感を呼び,主観的な視点を提供できるビジュアル・グラフィックスは次世代のゲームマシンの不可欠な要素技術とみなされるようになった。

こうしてプレーヤーのニーズは高度化し,その要求を満たすためにはもはや3次元データを扱うCG技術による他ない段階にまで達し,その結果として昨年からいよいよCGをその表示技術の基礎にしたゲームが出現した。ミッドナイトランニングである。これは3次元画像処理をして画面表示を行う画期的なものではあったが,面の塗りつぶしを行うとゲームの必須条件であるリアルタイム性が損なわれるため,黒い画面に点だけを描いた夜景であり,書換えにある程度時間をかけられるフライト・シミュレータとして作られた。

ミッドナイトランニングが出荷されたところ,大変好評で「ぜひ昼間のものを作ってくれ」といった要望が高かったそうである。そのため,さらに高度な画面表示を実現すべくゲーム専用のグラフィック・エンジンが開発されることになった。

こうして最新のゲームであるトップランディングやウイニングランが実現した。ウイニングランはスプライトをもたない新しいタイプの描画ハードウエアとして,「ポリゴナイザー」と名付けられたジオメトリ・エンジンをもっている。メインCPUは座標変換までを行い,専用のカスタム回路でポリゴンを描画する。このカスタム回路をポリゴナイザーと名付けたわけである。結果として最新のグラフィックス・スーパーワークステーション(GSWS)とよく似た構成のマシンに仕上がっている。

ゲームマシンのグラフィック・エンジン

グラフィック・エンジンとしては,描画専用チップを中心にカスタム回路をもった専用チップを組み合わせたものを独自開発している。またメインCPUとグラフィック・エンジンとはメモリを共有する密結合構造をしており,この点でも非常によく似ている。とはいってもGSWSはメーカーがニーズに先行して開発したものであり,価格的にもかなり高価なものである。対してCGゲームマシンは技術者が提案して作られたものではなく,コスト管理もきわめて厳しい。したがって,似ているとはいってもその中身は全く違った設計思想と実現内容をもっている。メインCPUのパフォーマンスの違い(最新のRISCチップvsオーソドックスなMC 68000)や,開発言語の違い(汎用性が高く移植性を重視した高級言語vsアセンブラ),バスの仕様の違い(汎用性・拡張性とネットワークを前提とした高速バスvsコストレスな専用クローズド・バス),グラフィック・エンジンの仕様(高速かつ汎用性を最大限要求され,標準グラフィック・パッケージの走るプラットフォームvs決まった形状を決まった条件で描画するファームウエア)などである。しかし,こうした最適化設計とパーツのコスト・パフォーマンス向上のため,現在の最新CGゲームマシンは特定のソフトウエア(専用のゲーム)を走らせるという条件では,5倍以上高価なGSWSとほぼ同等のポリゴン表示速度を達成している。

タイトーの「トップランディング」のゲーム中画面

3D-CG ゲームの実際

前置きが長くなったが，以下では実際の CG ゲームを紹介しよう。取り上げるのは，タイトーの「トップランディング」とナムコの「ウイニングラン」である。

■トップランディングについて

ゲームの形式はフライト・シミュレータであり，セミリアルタイムレスポンス（30 フレーム/秒は実現していない）である。ゲーム内容は離・着陸の操縦疑似体験で，ゲームをスタートすると最初の離陸の操作を行う。その後，日本各地の空港への着陸を繰り返し，うまく着陸できたらシーンが次の空港に変わり再び着陸を行う。視界は操縦席から見た空港周辺の空中からの景色になっており，プレーヤー操縦で見る角度が変わると新しい風景が見えてくる。つまり，常に現在見えているシーン以外のデータももっており，高速化のためにスクリーンに入るシーンのデータだけを選び出して計算表示するようになっている。座標変換の計算は専用 CPU（320 C 25）を使い，塗りつぶしは専用ハードウエアで実行する。それ以外の処理（得点計算や金が投入されたかどうかのチェックなど）はメイン CPU の MC 68000 で行う。MC 68000 と 320 C 25 はメモリを共有する構成になっている。モニターの表示解像力は 512×400，表示色数は 4,096 色，最近流行の体感ゲームであるため，ボディーソニックサウンドを有している。またバイクの走行シミュレーション・ゲームほどではないが，キャビンが動いて飛行機の姿勢をプレーヤーに伝えるようになっている。

高速化の手法は，①四角形のみに特化したジオメトリ・エンジンの設計，②データの削減（視線方向からの面しかデータがない。1 つの建物を 3 つの四角形ポリゴンだけで表示），③ポリゴンどうしの前後判定は各ポリゴンについては行わず，エリアブロックを設定しておいてブロックごとに前後判定する――などを行っている。

■ウイニングランについて

ゲームはフォーミュラ・カーレースで，コースや自分の車／他の車など，すべての物体は 3 次元ポリゴンで構成されている。視点は実際のフォーミュラ・マシンと同じ地上 80 cm であり，常にグローバルなデータの中で移動する固有のローカル視点座標から見た視野を表示する。バックもでき，コースを逆走することもできる（裏側のデータももっている）。バックミラーには追い付いてきた，あるいは追い越した他車が映る。運動計算はエンジンの出力特性，ギアのポジション，タイヤの摩擦係数，車重，空気抵抗，運動慣性など多くのフィジカルな物性をインプリメントしたシミュレーション原理で行われる（スリップ・ストリーム走行も可能）。

リアルタイムなレスポンスをもった画像を作り出すのはポリゴナイザーと名付けられたグラフィック・エンジンで，ゲーム専用のカスタム・ハードウエア・システムであり，1/60 秒ごと

ナムコの「ウイニングラン」のゲーム中画面

に1,000個のポリゴンを画面表示するパフォーマンスをもっている。

以上の仕様により，ウイニングランはこれまでのドライブゲームを超えたより主観的な体験ゲームに仕上がっている。具体的にみてみよう。5段変速のギアを使って，スピードとカーブに応じて加わる遠心力を感じながら限界ギリギリにドリフトさせてコーナーをクリアする手応えは初めてのものだ。これまでのドライブゲームでは実際のドライバーシートの位置から外の景色を眺めるものはなかった。ゲームマシンのスクリーンには自分の車が他の車や景色の手前に映っており，例えばスピンした場合などプレーヤーの目の前にある自分の車がくるくると回転する。景色や他の車はそれまでの運動を続けている。これはラジコンやスロットレーシングカーのリモコン感覚だ。ウイニングランでは遠近感も実際のスケール感覚に合ったものであり，その中を横方向の加速度(G)を感じながらコーナリングしているとテールが滑り始める。カウンター気味にステアリングを操作するとドリフトが始まり，きれいにコーナーをクリアできる。しかし，このときシフトミスをしたり急激にアクセルを開け過ぎると一気にスピンしてしまう。スピンすればまわりの風景が激しく回り，その中を他の車が追い越していく。バックギアにしてハンドルを切り返し，コースアウトした車をコースに戻し，再び走り始めると逆走することもできる。相対速度600kmですれ違う対向車はおそるべき迫力で迫り，脇をすり抜けていく。

こうした画面表示を実現するために，前述したポリゴナイザーによる高速ポリゴン描画機能を中心にシーンデータを生成するためのCIGシステム(Computor Image Generation System)を開発し，使用している。また，ナムコではこのCIGシステムと体感を与えるシステムとを合わせて「システム21」と名付け，今後のアーケードゲームの基本になる中心技術として位置付けている。

CGとゲームのかかわり

これまで述べてきたように，CGの応用分野にゲームが新しく加わった。これまでのCG技術の中では，リアルタイムな画像生成へのアプローチはそれほど活発には追求されなかった。しかし，ここでは基本的な条件にリアルタイムなレスポンスが要求され，かつ厳しいコストダウン要請があり，その枠の中で可能な限度のクオリティを模索する必要がある。従来以上に難しい制約だといえる。しかし，CG技術が普遍的で汎用な技術に育つために，こうした条件は当然克服される必要がある。要素技術はそれを包含する総合技術全体に制約を課するものであってはならないからだ。ゲームだけではなく今後多くの実務分野に用いられるであろうCG技術は，こうして多くの制約を次々と解決することで応用の利く技術として発達していくのであろう。建築などのプレゼンテーションにCGが応用され始めたと

ナムコの「ウイニングラン」ゲーム機全景

きに求められた条件として，適切な納期で必要とされるレベルのクリオティを実現し，リテイク(手直し製作)に耐えられる製作体制と生産システムを確立するというテーマがあったが，ゲームにCGを応用することが一般化すればまた別の切り口から生産性を高めるというノウハウが生まれるだろう。

とはいえ，CG技術の応用はまだ始まったばかりだ。各種の仕事の中にCGをいかに利用するかという応用技術の開発は，まだこれから本格化する問題である。CGを利用するゲームもさらに高度なものに進化するだろうから，今後そうしたCG以外の要請にこたえるかたちでCGの進歩が活発になり，袋小路に入りかけていたCGの発達が正常なものになることができればなによりだと考える。

ゲームの発達について

これまで述べなかったが，もともとゲームには具象的な内容を追求するものと抽象的な世界を演出するものがある。表示技術の発達はまずリアリティーを実現するため，こうした進歩は最初もっぱら環境型のゲームを進化させる。現実のビデオゲーム発達史をみても，記号的ゲームから具体的疑似体験ゲームへの流れが明確であろう。それを支えているのはマン・マシン・インタフェースの進歩であり，具体的にはリアルサウンド・システムによる聴覚，CGによるリアルな視覚，ボディーソニックや動きの伝達による体感などである。換言すれば，ゲームは環境そのものへの道を着実に進んでいるということでもある。もっとも，マン・マシン・インタフェースの進歩はいずれより高度な記号的象徴性を実現するであろうから，今後われわれを取り囲む環境は手に取ることのできる現実と虚構(仮想世界)の多重化されたリアリティーをもつことになるであろう。これは遊びのクリエーションあるいはアミューズメントの普遍化である。ゲームの中に存在する遊びやおしゃれ心，ゆとりのエッセンスは今後生活の中の要素技術として拡散し日常化していくであろう(例えば，からくり時計の流行やネオモダン建築の隆盛など)。

(近藤　和央)

Sunが画期的なWSを続々と発表の予定

ワークステーションのトップ・ベンダーであるサン・マイクロシステムズ社のスコット・マクニーリ社長らが語ったところによると，サンはこれまでの長い沈黙を破って，今年の第二四半期以降にきわめて意欲的な新製品を数多く発売することになりそうだ。

同社の新製品はMPU別にみると，モトローラ系，インテル系，SPARC系の3種類に分けられる。

モトローラのMC 68030を用いたワークステーションとしては，3.5 MIPSのローエンドGWSが1万ドルくらいの価格で出てくる見込み。同じくクロック周波数33.3 MHzの68030を用いた7 MIPSマシンが3万ドルくらいで第二四半期に発表される予定。また，インテルのi 80486を用いた10 MIPSマシンの年内発表も予定されている。

RISCアーキテクチャであるSPARCを用いたシリーズとしては3機種が予定されている。低価格なデスクトップ型は10〜13 MIPS，2.5 MFLOPS，デスクサイド・モデルが10〜15 MIPS，3 MFLOPSで2万ドル程度，上位のサーバー・モデルが15〜20 MIPS，4 MFLOPSで約4万ドルを予定している。参考までに，現状のSun-4/280は10MIPS，1MFLOPSである。

SPARCはScalable（拡張可能）の言葉通り，使用する回路設計・製造技術によって段階的に性能を向上させることが可能なアーキテクチャである。CMOSゲートアレイを用いた現在のSun-4は10 MIPSであるが，これをCMOSカスタムにしたものが10〜20 MIPSの能力をもつ。さらにECLバージョンでは35〜40 MIPS，10 MFLOPSというSPARCチップの発表が今年末から来年始めに予定されている。さらに，1991年から92年にかけては250 MIPSのガリウム砒素チップの発表も予定されている。

Sunワークステーションはこれまでグラフィックスに弱いとされてきたが，今後はグラフィックスも画期的に強化される予定。同社では24ビット（1,667万色同時表示）がワークステーションの標準になるとみており，ハイエンドなグラフィック性能としては現在の15万3次元ベクトル/秒，2万シェーディッド・ポリコン/秒から80〜100万3次元ベクトル/秒，15万シェーディッド・ポリゴン/秒になると推測されている。

低価格なデスクトップ型にも新しいチップを搭載した画期的なグラフィックスが採用される予定であり，13万3次元ベクトル／秒が見積もられている。また，これには4Mバイトのフロッピと富士通の100Mバイト・ディスクも予定されている。SunははっきりとNeXTワークステーションを意識しており，Sunがコンピュータ・ショップで買える日もそう遠いことではないであろう。

立石電機が低価格な2次元GWSを発表

パーソナルコンピュータをベースとしたパーソナルCADが非常に勢いづいて伸びているが，ソフトウエアが高機能化する一方で，ハードウエアの限界からグラフィック機能をもったワークステーションへの需要が高まっている。32ビット・パソコンやいわゆるUNIXパソコンもこれに呼応したものとも考えられるが，これらはグラフィック・プロセッサをもっておらず，CPUの負担なしにグラフィックの計算を行うことはできなかった。

こういった状況の中で，100万円を切るMC 68030ベースの低価格ワークステーションLUNAを出している立石電機が，標準構成でグラフィック機能をもちながら170万円という低価格を実現したGWSを投入する。

同社が今回発表したワークステーションは4機種で，このうちGWSの「グラフィックワークステーションLUNA-GW」にはデスクサイド型とデスクトップ型の2機種がある。両機種とも基本的に2次元GWSで，CAD/CAM/CAEが主な利用分野として想定されている。デスクサイド型GWSは，クロック周波数33.3 MHzのMC 68030をMPUに，ASICによるグラフィック・プロセッサを用いて60万ショート・ベクトル／秒を実現している。価格はハードディスク，ディスプレイなどを含めて約500万円となっている。デスクトップ型GWSは従来のLUNAのビットマップ・ディスプレイ・ボードを新開発のグラフィック・ボードに換えたもので，10万ショート・ベクトル／秒のグラフィック表示能力をもつ。このモデルでは20 MHzクロックのMC 68030が用いられている。グラフィック表示は1,670万色中256色同時表示可能で，価格は標準構成で170万円である。この他に1.3Gバイトまでのハードディスクを搭載可能なファイル・サーバー・モデルと，システムとして利用するためのディスクレス・モデルが発売となる。

周辺機器としては，LUNAが標準装備しているPCスロット（PC-9800シリーズの互換スロット）用アクセラレータ・ボードとしてトランスピュータが加わる他，NeXTなどで話題になり，ソニーのNEWSワークステーション用にも発表されているMOディスク（書換え可能な光磁気ディスク）も接続可能になる。

new products

3次元建築企画設計支援システム

ダイナウェア

ダイナウェアは，斜線制限，日陰・逆日陰も統合処理できる3次元建築企画設計支援システム「DYNA ARCH IPLAN」の販売を開始した。

同製品は，タブレットで地図をそのまま写し取ることにより簡単に敷地形状を入力できる建築用設計支援ソフトウエアである。また，入力された敷地の情報へ建物を建築する際の用途地域，容積率，建坪率，斜線制限，日陰規制などの法的な規制の入力が，法律知識がなくても簡単に行える。入力された敷地の建物が2階建て以上の高層構造をなしている場合，各階の法的な規制も自動計算される。計算結果は3次元のメッシュ型の等高線として画面に表示され，360°あらゆる角度から見ることができる。操作は，メニューとタブレットにより行える。

作成データは，3次元建築設計用ソフトウエアであるダイナパース3に読み込むこともできる。同社では，ダイナパース3とともに建築分野での需要を見込んでいる。

価格は100万円。PC-9801シリーズのMS-DOSで動作する(PC-98XL²はノーマルモードのみ)。

問合せ先：株式会社ダイナウェア
〠563　大阪府池田市石橋1-9-13
☎0727(62)8201

2次元図面から立体図作成

日本自動作図研究所

日本自動作図研究所は，すべての平面図を立体図に変える3次元図形処理システム「Koma View」の販売を開始した。

Koma Viewは，設計，配置図などの2次元図面から簡単に立体図を描くことができる3軸同時入力の図形処理システムである。立体図作成の経験がなくても作図が行えるという簡単な操作を実現しており，短時間のトレーニングで複雑なイラストも作成できるようになる。同社ではデモなども行っているため，実際に操作することができる。

価格は，パーソナルコンピュータ，X-Yプロッタを含めたシステムで300万円台となっている。

問合せ先：有限会社日本自動作図研究所
〠169　東京都新宿区高田馬場1-31-8
☎03(204)7090

ASICの論理設計CADシステム

富士通

富士通は，セミカスタムLSIにRAM，ROMを組み込む際の回路データを自動的に生成するソフトウエア「BANK-CAD」システムの販売を開始した。

これまでASIC(特定用途向けLSI)のセミカスタムLSIの中でゲートアレイやスタンダード・セルにRAMやROMのデータを組み込むために，同社の大型コンピュータやEWSを用いて回路データを生成していた。これでは客が自ら回路のシミュレーションを行うことが困難であり，能率が悪いことから，パーソナルコンピュータで実現できる論理設計用CADシステムが生まれた。

システムは，パーソナルコンピュータFMRシリーズで稼働し，パーソナルコンピュータ上で必要なセルをコンパイルすることにより，遅延時間を含んだシミュレーション・モデルを生成することができる。これにより，従来に比べて約1週間の設計期間の短縮が見込まれる。また，BANKCADで生成した回路データをシミュレーション用ソフトウエアVIEW-CADの入力データとして使用することができる。適用分野は，CMOS(相補型金属酸化膜半導体)のスタンダード・セル，ゲートアレイなどである。

価格は10万円。またSRAMジェネレータ，ROMジェネレータとも6万5,000円となっている。

問合せ先：富士通株式会社MOS事業部技術支援部
〠211　神奈川県川崎市中原区上小田中1015　☎044(777)1111

汎用2次元CAD/CAM/CAE

武藤工業

武藤工業は，汎用2次元CAD/CAM/CAEシステム「Expert Design VISION ver 4.0」の販売を開始した。

Expert Design VISIONは，2次元CADシステムEasy Draf²の上位機種として位置付けられる製品であり，Easy Draf²と同じくスイスのベンチャー企業AELアドバンスト・グラフィック・システムズ社との技術提携により完成した。同システムは，自社ノウハウを組み込んだオリジナルCADシステムの構築，および応答性・操作性への要求が高まっていることに対応している。

増大する一方の図面ファイルを体系的・階層的に管理でき，ワイルドカードによる検索も可能であり，操作中の不明な点に関しては日本語オンライン・マニュアルを画面上に表示することができる。さらに，個々のユーザーごとのノウハウをプログラム化してユーザー・コマンドを作成したり，オペレーション・フローをシステムに組み込み自動設計を支援する中間言語がサポートされている。

機能的には，表示されている画面の要素のデータを取り出してその値を変化させ，その結果を即座に表示するエンジニアリング・スプレッドシート，パラメータや設計仕様値を指定するのみで自動的に図面を作成するパラメトリック・モデリング，1つの画面に複数のファイルが呼び出せるマルチファイル/マルチウィンドなどの機能が装備されている。操作は極力自動化されており，キーを使う回数が少なくてすむ。

価格は，ホスト・コンピュータ，主記

憶 8M バイト，130M バイト固定ディスク装備の HP9000 シリーズを採用した基本システムで 800 万円から。システムにはプリンタ，タブレット，テープドライブが付属される。オプションでプロッタの接続も可能である。

問合せ先：武藤工業株式会社　総合企画室　〒154　東京都目黒区中目黒 4-6-1　☎03(760)6111

省操作の汎用パーソナル CAD

アンドール

アンドールは，同社のパーソナル CAD の設計思想である“省操作”をさらに推し進めた汎用 CAD「CADSUPER SX III」の販売を開始した。

同社ではパーソナル CAD の設計思想として，少ない操作でたくさんの作業を行えることを目的としているが，このコンセプトのもとに新製品が発表された。同製品は，32 ビット・パーソナルコンピュータをベースとした CAD システムであり，マルチウィンド上でマクロキーやマクロコマンドを使用して操作できる。マクロキーは空いているキーに対して割り付けることができ，それらは簡単に呼び出せる。

機能的には，部品などの指定した部分を複写・移動させて他の部分との干渉をチェックするドラッギング・シミュレーション機能，前もって指定していた図面部分をワンタッチで拡大・縮小するマルチズーミング，パラメータを入力して図形を作成するパラメトリック機能などが装備されている。また，最大 3 万要素の大きな図面データをメモリ上で操作するため高速な作業が可能である。

価格は 138 万円。PC-9801，FMR，パナコムなどのパーソナルコンピュータで稼働する。

問合せ先：アンドール株式会社 CAE システム部　〒650 兵庫県神戸市中央区中町通 2-3-2　☎078(351)5241

400dpi の A4 イメージスキャナ

シャープ

シャープは，A4 サイズで最高 400dpi のイメージスキャナ「JX-40」の販売を開始した。

JX-400 は，A4 サイズの原稿を最高 400dpi(約 15.7 ドット/mm)の高解像度

で読み取ることができる。さらに，30～400dpi の範囲で解像度の指定ができ，0.04 インチの間隔で読取り範囲指定ができる。これらの機能を組み合わせることによって拡大・縮小・レイアウトなどを行うことができる。

読取り階調は，256/16/4 階調または二値化の指定をすることができる。中間色のある原稿も，疑似階調処理として組織ディザ法と誤差拡散法を用いているため最適な状態で読み取れる。組織ディザ法は，マトリクスを設定してマトリクス内で疑似的に階調を表現する方法であり，誤差拡散法は過去の変換結果に基づいてドットの差を動的に変換する方法である。また，読み取った後は MTF(変調伝送関数)補正によってエッジを改善し，画像の鋭さを向上させる。さらに出力の際はガンマ補正を行って各出力デバイスに合わせた濃度に設定できる。実際の読取り時間は約 10 秒となっている。インタフェースは GP-IB。

価格は 29 万 8,000 円である。

問合せ先：シャープ株式会社 情報システム事業本部第 2 商品企画部　〒639-11　奈良県大和郡山市美濃庄町 492　☎07435(3)5521

電算写植機への
コンバート・システム

東洋産業

東洋産業は，キヤノンの DTP システム EZPS のデータを写研の電算写植システムのデータコード SK72 に変換するシステム「EZFIT SK レベル 1」の販売を開始した。

同システムは単に文字コードの変換を行うのみではなく，書体と文字のサイズ，用紙サイズや用紙方向などの用紙書式，組方向・段数・段高などの版面書式，柱，ノンブル，アンダーライン・囲み罫・ルビ・上付き・下付きなどの各コマンド，それに大見出し・中見出し・小見出しなどの各見出しの変換もできる。EZFIT の指定で変換できないものは，変形文字，白抜き文字，回転文字などであり，大部分のデータはそのままのデータとして電算写植で使える。

同社ではすでに電算写植機にもデータ出力できる DTP システム Linernote シリーズを販売しており，今回のデータ・コンバータはそのノウハウを生かして開発されたものである。

価格は，ソフトウエアのみで 118 万円，PC-9801VX41 を用いたトータルシステムは 214 万 9,000 円，PC-9801RA5 を用いた場合は 234 万 9,000 円となっている。EZFIT を使用する場合は EZPS の拡張 MS-DOS コンバータが必要である。

問合せ先：東洋産業株式会社　〒101　東京都千代田区神田須田町 2-25　☎03(256)7355

100 万円を切る大型プロッタ

グラフテック

グラフテックは，100 万円を切る低価格機「大型パーソナルプロッタ 7000 シリーズ」2 機種の販売を開始した。

A1 サイズの「FP7100」と A2 サイズの「FP7200」の 2 機種が販売された。それぞれ，低価格の実現とともに専用の 16 ビット CPU をもたせ，高速処理が可能になった。専用の 16 ビット CPU の内蔵により，最大速度 600mm/秒の高速作図が可能になった。メカニカル解像度は 0.005 mm で，同時にペン・ソーティングやベクトル・ソーティングなどのインテリジェント機能を発揮する。

操作は LED パネルを見ながら対話的に行い，約 60°の傾斜でも使用が可能。ペンは水性，油性，鉛筆，ボールペンなど，8 本使用できる。標準で 40K バイト，オプションで 1M バイトのバッファメモリを内蔵しているため，一度に大量のデータが送れる。ホスト・コンピュータとのインタフェースは RS-232C またはセントロニクスに対応している。

価格は，FP7100 が 79 万円，FP7200 が 64 万円，オプションの 1M バイト・バッファメモリは 6 万円。

問合せ先：グラフテック株式会社 プロッタ事業部営業部　〒252　神奈川県藤沢市遠藤 1937　☎0466(87)4111

new products

MS-DOS ファイルを読取り可能な MacII

アップルコンピュータ

米国アップルコンピュータ社は，同社のパーソナルコンピュータ Macintosh シリーズの新製品として，32 ビットの「Macintosh II cx」を発表した。

CPU は従来の Mac II x などと同じ 16MHz の MC68030 を採用しているが，従来製品に比べて大きさがコンパクトなものになっている。また，従来の Mac と異なりスーパードライブとよばれるディスクドライブを採用したため，MS-DOS など Mac 以外のフォーマットのディスクを読むことが可能になった。ソフトフエアは従来製品と互換性がある。他に，新規の機能として停電でシャットダウンした後，自動的にシステムが立ち上がるオートリスタート機能などがある。現在，日本語化については検討中である。

価格は，固定ディスクなし，主記憶 1M バイトの基本タイプで 4,669 ドル。

問合せ先：アップル コンピュータ ジャパン株式会社 マーケティング部
〒106 東京都港区六本木 1-4-30
☎ 03(224)7000

Macintosh 用 CAD ソフト

システムソフト

システムソフトは，米国 Claris 社の Macintosh 用 CAD ソフトウエア「ClarisCAD(英語版)」の販売を開始した。

ClarisCAD は，Mac 用に開発された建築，建設(AEC)，機械などのエンジニアリング業務や，工業デザイン・製図といった多目的デザインのために開発された CAD ソフトウエアである。使用ユーザーとして初心者も対象としているため，操作が簡単である。また，付属の学習専用ビデオテープを用いれば 45 分でその操作方法が修得できる。また，テクニカル・ユーザーが図面を作成する際も通常の CAD デザイン時間の 1/2 の時間で作成できる。

編集できる線は，輪郭線，丸溝線，ハッチング，二重線など多数が用意されており，設定によって，端線，線と線との交差，中心点，接点，角，垂線，接線および分岐点に関してスピードアップを図ることができる。部品やシンボルはライブラリ化して使用することができる。また，部品は MacDraw IIのデータを利用できる。

作成したデータは，ImageWriter，LaserWriter の他，Mac につなげる PostScript 対応のプリンタや HP 社などの製図用プロッタに出力できる。また，DXF や IGES などのデータに変換することもできる。

価格は 12 万 8,000 円。使用するには固定ディスクと System 6.0 以上の OS が必要である。

問合せ先：株式会社システムソフト
〒810 福岡市中央区天神 5-7-2
☎ 092(714)6236

CD-ROM 標準搭載の 32 ビット・マルチメディア・パソコン

富士通

富士通は，低価格の 32 ビット・マルチメディア・パーソナルコンピュータ「FM TOWNS」1 機種 2 モデルの販売を開始した。

マルチメディアとは，文字やデータなどの他に音声や音楽の世界，絵や写真の世界を情報として同時に扱えることで，Macintosh や X68000 ではすでに実現されている。FM TOWNS は，パーソナルコンピュータで初めて標準で 540M バイトの CD-ROM ドライブを装備しており，大量の画像データ，音声データなどを扱うことができる。CD-ROM は，グラフィック・オーディオの規格を追加した CD-ROM 拡張フォーマット XA ＊1 にも将来，対応する予定。オーディオ機能としては，FM 音源＋PCM 音源＋CD デジタルオーディオ再生により高音質な音楽・音声を再現できる。ビジュアル機能では，専用の高速スプライト機能や，1,677 万色中 256 色（640×480 ドット），32,000 色(320×240 ドット)などのサポートにより，自然な色調の高速なグラフィック画面を表示できる。

CPU には i80386 を採用し，OS には 32 ビット本来のパワーを発揮する i80386 ネイティブモードを直接操作できる専用の TOWN OS を採用した。PC-9801RA の MS-DOS や OS/2 の 16 ビット用 OS などと異なり，本来の 32 ビット・アーキテクチャを実現した。OS は，全操作を画面上のアイコンを選ぶだけのマウス操作を基本としているので，難しいコマンドを知らなくても操作できる。

ソフトウエアとして，絵・音楽・音声・文字を自由自在に統合できるハイパーメディア対応のクリエーティブツール TownsGEAR や，3 次元ゲーム AFTER BURNER などで約 100 本の CD-ROM ソフトを年内に提供する予定。さらに，別売の専用日本語 MS-DOS Ver.3.1(MS-DOS エミュレータ)を利用すれば，Lotus1-2-3，一太郎など FM R-50 シリーズ上で動くソフトウエアを活用することができる。富士通ワードプロセッサの FM-OASYS Ver.1.0 も本年秋に提供する予定である。

価格は FM TOWNS モデル 1 が 33 万 8,000 円，FM TOWNS モデル 2 が 39 万 8,000 円となっている。

同社では，今回の FM TOWNS の発売にあたり，強力な販売プロモーションを展開していく予定である。また，FM TOWNS の発表と同時に，米国ルーカスフィルム社と提携してマルチプレーヤ通信ゲームソフト「HABITAT」の技術契約を結んだ。これによりパソコン通信サービス NIFTY-Serve で，先進的なゲームを提供する。とりあえず FM TOMNS 用を作成する予定という。

問合せ先：富士通株式会社
〒100 東京都千代田区丸の内 1-6-1
☎ 03(216)3211

サーバー版 TITAN

クボタコンピュータ

クボタコンピュータは，グラフィック・スーパーワークステーション TITAN のサーバー版「TITAN サーバー」の販売を開始した。

TITAN は，MIPS 社の RISC プロセッサ R2000 を CPU に採用した並列演算型のグラフィック・スーパーワークステーションであり，特徴としてベクトル・ユニットによる高速グラフィック描画能

力があげられる。今回発表されたサーバー版は，グラフィック機能およびその周辺装置を除いた演算専用のコンピュータとして低価格化を図ったものである。同じ価格帯のミニコンピュータに比べて5〜10倍の価格性能比を有している。

ハードウエアは，1〜4プロセッサ，16M〜128Mバイト主記憶，380M〜2280Mバイト固定ディスク，最大10Gバイトの外部記憶装置などで構成される。OSは，UNIX System V＋4.3BSD，最適化Cコンパイラ，最適化FORTRANコンパイラが使用できる。整数演算性能は1プロセッサで16MIPS，4プロセッサで64MIPS，浮動小数点性能は16(1プロセッサ)〜64MFLOPS(4プロセッサ)である。

同社では，適用分野として分子設計，構造解析，金融シミュレーションなどを見込んでいる。

問合せ先：クボタコンピュータ株式会社 企画室
〒160 東京都新宿区新宿2-8-8
☎03(225)0931

X68000シリーズを拡充

シャープ

シャープは，同社が販売しているパーソナルワークステーションX68000の新機種として，主記憶2Mバイトを標準装備したシリーズの最上位機種「X68000 EXPERTシリーズ」と，標準機種「X68000 PROシリーズ」の販売を開始した。

両シリーズとも，CPUは従来と同じ16ビットMC68000を採用しているが，標準で65,536色同時表示可能，スプライト機能，スーパーインポーズ機能，FM音源，音声合成機能など，ゲーム，テレビ，オーディオを意識したマルチ機能を継承している。新製品では，主記憶2Mバイトの標準実装(EXPERT)や40Mバイト固定ディスク内蔵などユーザーの使用範囲が拡張されている。また，拡張スロットとして4スロット内蔵されているため，MIDIボード，増設RAMボード，FAXボード，数値演算プロセッサ・ボードなどが内蔵できる。

OSは従来と同じ同社オリジナルのHuman68kのVer.2.0を搭載。マルチウィンド上でマルチタスク的な処理を行うことができる。従来のX68000シリーズのソフトウエアをそのまま使用できる。また，ファイルアクセス時間は2倍にスピードアップし，UNIX的なヒストリ機能，alias機能，バッチ機能などを備えている。漢字入力のための日本語フロント・プロセッサは標準装備している。

価格は，主記憶2MバイトのEXPERTシリーズがCZ-612C(40Mバイト固定ディスク内蔵型)が46万6,000円，CZ-602C(固定ディスクなし)が35万6,000円。主記憶1MバイトのPROシリーズのCZ-662C（固定ディスク内蔵)が40万8,000円とCZ-652C(固定ディスクなし)が29万8,000円となっている。

問合せ先：シャープ株式会社 電子機器事業本部テレビ事業部第4商品企画部
〒329-21 栃木県矢板市早川町174
☎0287(43)1131

サイクロンがバージョンアップ

アンス・コンサルタンツ

アンス・コンサルタンツは，レイ・トレーシング・ソフトウエア「サイクロン」のバージョンアップを行うと発表した。また，同時にサイクロン用のアニメーション作成ツール「アニメキット」，サイクロン開発ツール「サイクロンAART」の販売を開始する。

サイクロンは，PC-9801，X68000などのパーソナルコンピュータで動くレイ・トレーシング・レンダリング・ソフトウエアであり，レンダリングの機能以外にもモデリングの機能をもっている。今回のバージョンアップでは，機能面ではモデラー部分にポリゴン・モデラーが追加され，レンダリング部分ではアンチ・エイリアシング，光源ファイルのコピーなどの機能が追加される予定である。また，性能面では，レンダリングにボクセル分割法を採用し大幅な演算速度の向上が実現される。同社によると，同じパーソナルコンピュータ用レイ・トレーシング・ソフトウエアC-TRACEで10日かかる計算が34分で実現できる。

サイクロンAARTは，ソースプログラムを公開したものである。これは最近要求されている3次元CGとCADとのデータ交換に対応したものである。自社でレイ・トレーシングのアルゴリズムの研究，設計・開発を最初から行うには膨大な開発コストが必要とされることから，レンダリング部分のみのソースを公開した。これにより，非常に少ない開発コストで，レイ・トレーシングのアルゴリズムを自社製のソフトウエアに組み込んでデータ交換に使用することができるようになる。記述言語はLattice C ver 3.2。

アニメキットは，サイクロンでアニメーション作成を行う際のソフトウエアであり，16コマ/秒のアニメーションを作成することもできる。ダイナミックムーブ機能とよばれる動画機能を使い，サイクロンで作成した3次元画像を回転，移動などをさせることができる。

サイクロンAARTとアニメキットはサイクロン本体がないと動作しない。

価格は，サイクロンAARTが39万円，アニメキットが5,000円となっている。サイクロン本体のバージョンアップは無料である。

問合せ先：株式会社アンス・コンサルタンツ 〒810 福岡市中央区平丘町68
☎092(522)6347

超高速A/D, D/Aコンバータ

アナログ・デバイセズ

アナログ・デバイセズは，超高速の12ビットA/D, D/Aコンバータ「AD9005」の販売を開始した。

AD9005は，トラック・ホールド回路，電圧リファレンス・タイミング回路を内蔵し，アナログ信号のデジタル処理に必要な機能を小型46ピン・ハイブリッド(アナログ/デジタル)DIPに備えている。変換は独自のサブレンジ変換方式により，サンプルレートが10MSPS(MHzサンプリング/秒)の超高速変換が可能である。S/N比64dB，高周波歪み−72dBcである。

同社では，レーザー・デジタル・オシロスコープ，医療用スキャナ，高速通信などへの需要を見込んでいる。

価格は25万6,300円。

問合せ先：アナログ・デバイセズ株式会社 マーケティング部企画課
〒102 東京都千代田区麹町4-7-8
☎03(263)4788

PC-98用グラフィック・ライブラリ バージョンアップ

イシガキ エム・イー・エス

イシガキ エム・イー・エスは，PC-9801用のグラフィック・ライブラリ「超高速グラフィックユーティリティ High Level Graphics Ver 2.1」のバージョンアップを行った。

HGXは，OS(MS-DOS)に対してグラフィック機能をもたせると同時に，PC-9801独自のグラフィック・プロセッサGDCの機能を直接使用することにより，高速に描画を行うグラフィック・ライブラリである。今回，ノーマルモード用HGX on98とノーマル/高解像度モード用HGX on XX両方のバージョンアップが行われた。今回のバージョンアップでは，PC-98シリーズ全機種に対応するとともに，ライブラリアクセス言語が増やされた。

PC-98シリーズのほとんどの機種で使用可能であるが，E/F/Mのみ拡張機能の使用ができない。使用可能な言語はMS C，Lattice C，MS Pascal，MS FORTRAN，MS BASICなどのコンパイラ言語の他，dBASE IIIなどの簡易言語となっている。

バージョンアップ価格は無償となっている。新規に購入する場合は，HGX on 98が3万8,000円，HGX on XXが5万8,000円となっている。

問合せ先：株式会社イシガキ エム・イー・エス
〒130　東京都墨田区両国2-20-12
☎03(633)1761

MEGATEK社製GWSを販売

CRC

センチュリリサーチセンタは，昨年，米国MEGATEK社と日本における総販売代理店契約を結んだが，今回第1回製品として「Sigma20」の販売を開始した。

Sigma20は，サン・マイクロシステムズ社のSunシリーズをベースとして，MEGATEK社がグラフィック機能を追加したグラフィック・ワークステーション(GWS)。追加されたグラフィック機能は，OHC，TT，ICPなどのグラフィック・エンジンとグラフィック・ソフトウエアである。

OHCはフレーム・バッファを搭載しているグラフィック・ボード，TTはディスプレイ・リストメモリを搭載，ICPは入力制御を行うプロセッサであり，グラフィック描画はCPUではなくこの専用プロセッサが行う。グラフィック専用ライブラリとしては，ACCESS/20，EXPLORE/20が用意されている。基本的にSunのCPUは通常のアプリケーションなどに使用するように勧めているが，Sunの標準のグラフィック・ライブラリであるSun COREやCGIなども搭載している。そのため，Sunで開発された専用のグラフィック・アプリケーションの実行やSun用のグラフィック・アプリケーションの作成も可能である。専用グラフィック・エンジンを使用した際のグラフィック能力は，2次元ベクトルが30万本/秒，3次元ベクトルが20万本/秒，グロー・シェーディングを用いた3次元ポリゴン塗りつぶしが2万個/秒となっている。Zバッファは16ビット，同時表示可能な色は1,670万色となっている。

価格は，主記憶8Mバイト，固定ディスク288Mバイトの標準型にグラフィック機能をフルセット装備して，Sun-3ベースのモデル3060が1,200万円，同モデル3200が1,700万円，Sun-4ベースのモデル4100が1,700万円，同モデル4200が1,900万円となっている。

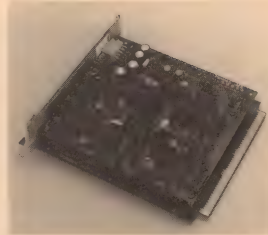
問合せ先：センチュリ リサーチ センタ株式会社 営業開発第2部
〒103　東京都中央区日本橋本町3-6-2
☎03(665)9828

PC-9800シリーズ用 NTSCフレーム・バッファ

メイコー・マルチアート

メイコー・マルチアートは，PC-9800シリーズ用の画像処理用フルカラー・フレーム・バッファ・ボート「FBN98」の販売を開始した。

同製品は，研究所やデザイン事務所でのフルカラーの画像処理用に作られたものである。通常，画像処理は白黒や少ないカラーで行っているが，これを使うことによりフルカラーでの解析が行える。画像処理用に作られたものであるため，このフレーム・バッファ用ペイント・ソフトウエアやCG用ソフトウエアは提供されない。メモリは1,024×512×3ピクセルでR，G，B各8ビット(約1.5Mバイト)である。

また，シンクジェネレータを搭載していることから，内部・外部同期のビデオ入出力も可能である。パーソナルコンピ

ュータからのアクセスはDMA可能なメモリマップ・バンク方式で，配列は処理しやすいピクセルマップを採用。メモリアクセスおよびデモ用のサンプルソフトウエアが付いてくる。

価格は36万円となっている。

問合せ先：株式会社メイコー・マルチアート 画像処理事業部
〒150　東京都渋谷区恵比寿1-19-15
☎03(447)8737

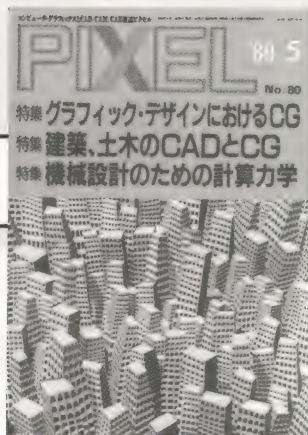
A3プロッタの新機種

グラフテック

グラフテックは，A3プロッタの新機種として「新・マイプロット4000シリーズ」の販売を開始する。

新シリーズは，A3サイズ小型プロッタのマイプロットシリーズを新しくしたもので，低価格・高性能を実現した。販売が開始されたのは，専用のエンジンであるデジタルサーボDS搭載の「MP4300」「MP4400」と，普及型の「MP4100」「MP4200」の4機種。デジタルサーボは，640mm/秒の描画速度と0.005mmの高分解能を有し，このクラスでは最高の高速・高画質を実現した。DSを作動させるのに専用の16ビットCPUを使用しているため，ペン・ソーティングなどのインテリジェント機能も実現した。また，40Kバイトのバッファメモリを装備，ホスト・コンピュータから大量のデータが受信できる。両方ともJIS第1/第2水準漢字，記号など約6,900文字を内蔵している。MP4300とMP4400の操作は，LCDパネルを見ながらの対話型で行うことができる。

価格は，1Mバイト FD内蔵，16ビットのMP4400が26万8,000円，MP4300が19万8,000円，8ビットCPUで静電吸着式のMP4200が14万8,000円，マグネット方式のMP4100が12万8,000円となっている。

問合せ先：グラフテック株式会社 プロッタ事業部営業部
〒252　神奈川県藤沢市遠藤1937
☎0466(87)2221

文房具感覚で日常性のCADとコンピュータ・グラフィックス

From The Editor's Desk

PIXELでは高度な3次元コンピュータ・グラフィックスやCAD/CAMの紹介が多いが，それは現在の日常性を離れた何か新しいものを紹介したいという編集者の性であって，それらが現在誰にでも求められているか，現在のニーズやコストに適合しているかどうかとは別である。ワークステーションにしても，ハイエンドで高価な3次元ワークステーションが誰にでも必要であるということはない。

最近，グラフィックスやCADを簡単に自分のものにしてしまおうという雰囲気が高まっている。3次元のグラフィックスをやるんだという思い詰めた感じではなく，文房具的に，本当に紙の上にサラサラと絵でメモをとるような感じで利用する。もちろん，2次元でいいし，簡単に色までつけてしまおう。また，簡単なCADシステムを使えば，立派な図形データができてしまう。技術文書の中の絵や打合せ用のスケッチ，カラーのプレゼンテーションが簡単に作れる。

これだけCADやグラフィックスが身近になってきたのだから，もっと手軽にコンピュータで絵を描けばいいと思う。PIXELでは難しい話ばかり掲載するので，コンピュータで絵を描くことを業務の中に限定したり，特定の専門家でなければ使えないような印象を与え続けてきた罪は確かにあろうと反省している。

日常性のCADやグラフィックス，文房具感覚のCADやグラフィックスがもっと使われていい。超高解像度であったり，フルカラーであったりは必ずしも要らない。とにかく，すぐに使える道具であることが第一だ。DTPにしても，研究者の間では論文作成用にTeXの利用が常識になりつつある。これも組版の専門家のためのDTPではなく，組版については素人の日常性のDTPである。

その場合のコンピュータについても高級なワークステーションは要らない。あっても問題はないのだが，高価だから誰にでも使えるというものではないので，パーソナルコンピュータでいい。贅沢を言わせてもらえるなら，現在のMS-DOSのパーソナルコンピュータは遠慮したいとかねてから思っているだけである。ユーザー・インタフェースを考えるとネットワークや広いメモリ空間，マルチジョブやマルチウィンドなどはやはり欲しい。それにもう少し解像度の高いモニターが欲しい。そのようなニーズに合っているのであろうか，マッキントッシュの人気が急速に高まってきている。あのハイパーカードは一度使ったら離せないであろう。ただし，アップル社のパーソナルコンピュータは日本では価格が高いのが難である。日本のメーカーに私の求めているパーソナルコンピュータの製品化を期待している。

1989年5月1日

河内隆幸

特集1 グラフィック・デザインにおけるCG

コンピュータ・グラフィックスがグラフィック・デザインの分野にどんどん進出している。印刷原稿のためのコンピュータ・グラフィックス画像の制作だけから，最近ではパーソナルコンピュータ上で最終完成原稿まで作成してしまうというDTPの流れをくんだ印刷のプリ／プレス処理までが行われ始めている。

コンピュータ・グラフィックスならではの豊富な色彩を利用したカラー・シミュレーションの容易さは，デザインツールとしてのデザイナーの作業ばかりではなく，CGに対する意識まで改革しつつあり，今やデザインの現場においてなくてはならないものという認識がされつつある。グラフィック・デザイン分野におけるコンピュータ・グラフィックス応用の広がりは，グラフィック・デザインという分野をも変革してしまう可能性がある。今後もコンピュータ・グラフィックスの非常に重要なポジションを占めるものとして，需要の拡大をみるうえでも大変注目される。

また最近では，操作性に優れたCGシステムの登場により，コンピュータ・グラフィックスは一部の特別な人でなくても利用できるようになりつつある。デザイン分野で，エアブラシなどと同様の一つのツールとして積極的に利用している電脳デザイナーが増えている。PIXEL 4月号のCGグランプリ速報でも紹介したように，その作品のクォリティの高さは目をみはるものがある。投稿作品の数の多さなどからみてもデザイン分野におけるその潜在需要の大きさは，コンピュータ・グラフィックス第二世代ともいえるような大きな広がりが期待される。この特集では，グラフィック・デザインの分野においてコンピュータ・グラフィックスがどのように利用されているか，またデザインにコンピュータ・グラフィックスを利用するためには何が必要なのかを考える。

今回の特集で紹介する例だけがグラフィック・デザイン分野でのコンピュータ・グラフィックスではない。デザイナーの意識変革が起こることによって，またどんな発想をするCGの異才が現れるやしれない。今後が楽しみである。

PIXELでは今後シリーズ企画として，グラフィック・デザイン分野におけるコンピュータ・グラフィックスの最新の話題をお届けする予定である。ご期待下さい。

●グラフィック・デザインにおけるCGのあり方

デザイン事務所におけるパソコンCGの活用事例として，具体例をあげながらデザイン作業におけるCGのあり方を紹介する。

● DAI・MEDIA

グラフィック・ペイントボックス，デジタル（ビデオ）ペイントボックス，ミラージュをメインとして，クリエータ自身が創作をし，最終原稿まで作成できるアーチストのためのデザインツールを紹介する。

●グラフィック・デザイナーのためのデジタル ペインティング入門

デザイナーの作業ばかりでなく，意識をも改革しつつあるCGシステムといかに付き合ったらよいか，CGが本当によくわかる数少ない現役デザイナーがそのコツを伝授する。

●パソコンによるプリ・プレス処理

パソコンによる印刷のプリ・プレス処理がDTPの普及に伴い，話題になっている。実際に印刷の現場ではどのように取り扱われているのか，CGと印刷に精通した著者がやさしく解説する。

グラフィック・デザインにおけるCGのあり方

具体的な制作過程を通して，グラフィック・デザインの中のCGの占める位置を，デザイン事務所の立場から考える。

伊藤　雅俊*

◉図1はCOLOR IMAGES参照，表1は次ページ。

はじめに

コンピュータ・グラフィックス（CG）をグラフィック・デザインで使用する場合，単にプレゼンテーションのツールとして使用する場合と，2次元，3次元の作画を通して総合的なデザインツールとして使用する場合に分けられる。前者は主にプレゼンテーションの省力化が目的であり，後者はより新しいビジュアル表現の開発が目的である。

当社のCGは約2年前，新しいデザイン開発の先行投資として導入された。当時は日本のパソコンCGの黎明期であり，いっせいに低価格CGシステムが発表された。パソコンCGは，それ自体低価格であり，周辺機器が充実していること，プログラム記述においても最近のC言語ブームの影響でかなりのレベルまでは資料がそろうことなどにより，普及の条件は整ってきている。また，最近のCPUの高性能化により以前の大型機に比肩するほどのキャパシティをもってきており，パソコンCGの普及とともに本格的CGの末端への広がりは，今まさに爆発寸前といった様相を呈している。

本稿では，具体的なグラフィック・デザインの制作過程を通して，レイ・トレーシングの魅力に迫ってみたいと思う。

■制作環境

(1)　ハードウエア

- CPU(PC-9801 VX 4＋TURBO-386＋80387)
- フレーム・バッファ(image maker，1,024×512×4)
- パソコン・モニター(14インチ PC-KD 854)
- RGBモニター(20インチ VM-300)
- 日本語プリンタ(PC-PR 101 F 2)
- イメージ・スキャナ(JX-200)
- タブレット(HDG 1111 B)
- RAM-DISK(4Mバイト)
- 増設ハードディスク(20Mバイト)
- カメラ(プレゼンテーションのための管面撮影用)

(2)　ソフトウエア

- MS-DOS Ver.2.11(Microsoft)
- vi-EDIT スクリーン エディター(アスキー)
- RAY-TREK II(ヴィーアイ)
- 3D animation system(ナムコ)
- 2D paint system(ナムコ)
- C-Compiler(各種)
- 自社開発画像処理ソフト

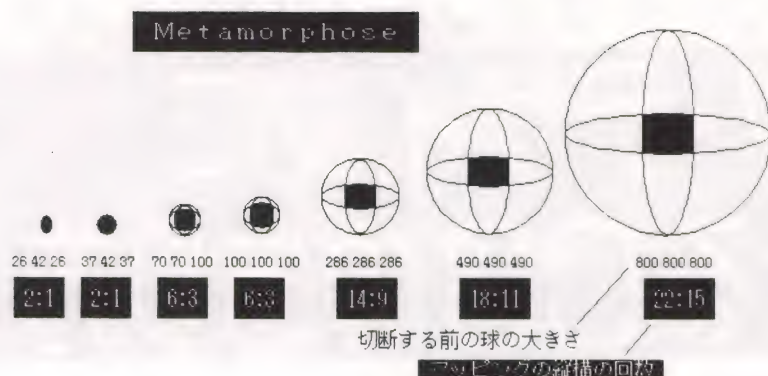
“Personal AS”のポスター（図1）

この作品のデータリストを**表1**に示し，制作順に解説を加える。

①クライアントとの全体的なコンセンサスをもとに，ディレクターとCGデザイナーとの間で全体のコンセプトをまとめる。この際CGデザイナーは，CGをいかに効果的に作品に反映させるかを突き詰めなければいけない。

イメージを固める過程には2つのタイプがある。1つは既成のCG作品から発展させる場合で，もう1つはラフスケッチのイメージを視覚化させる場合である。今回の場合は後者の方で，“新しい技術の創造”というコンセプトをもとに作られた。

当初，スキャンラインで制作する方針だったが，マッピン

*いとう　まさとし　㈱エースデザイン　CG　〒112　東京都文京区後楽2-2-10

◀図 2

●図3は78ページ，図4はCOLOR IMAGES参照。

グ能力，質感表現能力においてRAY-TREK IIの方が数段勝ることがわかり変更した。

②文字位置，配色などを考慮して3次元での構成を決定する。この際，各パートの形状がモデリング可能かどうか判断し，かつ，メモリ容量，計算時間について現実的選択をする。些細な形状，質感のこだわりがその構想の全体の断念につながらないようにする。

この作品の場合，卵からPC画面へのメタモルフォーゼで長径42，短径26の楕円体の全体から半径800の球の一部への形の変化と同時に，黒い画面をマッピングされた楕円体の反射率，拡散率の変化で黄金色からPC画面の黒への移行を実現している。各パーツの制作過程で画面の4隅を丸くしてほしいという要望があったが，全部の変化過程の画面を丸めることによる制作上の負担とかえってデザインの面白さがなくなるという判断から，最終画面のみ角を丸めることに落ち着いた。

③各パーツを全体の座標系に入れ，視界を決定し，低解像度，質感デフォルトで計算する。ここでは X，Y，Z それぞれの方向からレンダリングして，形の重なり，前後，上下関係などをチェックする。

④同じ座標系に球など計算の負荷の少ないダミーを置き，質感の徹底的なチェックをする。静止画のCGの場合，形よりも質感の方が大切になる場合が多い。

⑤全体をRGB画面大の解像度（640×480）でレンダリングして，最終調整する。この時点では全体の色調やバランスのチェックをし，大きな変更点があれば思い切って初期段階に戻って作業する。

CPUの能力が向上すれば計算の余裕がでて，始めからすべての要素を入れて計算した方がよいと思いがちであるが，CGデザインは建築や絵画と同じで，それぞれの作業を抽象化していくことが作品の質の向上につながる。形状をチェックしている段階で色はかえって邪魔である場合が多い。

⑥解像度を印刷に適した値に上げ最終レンダリングする。印刷の場合，A4判の仕上がりで1,024×1,024以上のポスターなどは最低でも1,536×1,536以上必要である。

⑦最終段階としてポジフィルムにしなければいけない。現在のところ，F社のレーザー出力が比較的高画質を得られるが，ピクセル間がシャープなため，低解像度の出力には不向きである。1,024×1,024程度であればI社の出力サービスも悪くない。

作品の出力は全体のイメージを左右するので，できるだけ高画質出力を選択すべきである。フィルム出力の段階で全体が暗く仕上がることと，マゼンタ系が強くなる傾向があるので，データの段階で考慮に入れてデザインした方がよい。

図2は画面の切断の仕方を説明したもので，卵形から画面への変化がよくわかると思う。

図3は切断前の球の様子をレンダリングしたもので，このマッピングの指定ができなければ今回の作品は生まれなかった。

林檜の木の作品(図4)

Cの入門書などで再帰関数を用いた木構造のプログラムがあるが，これを3次元で表すことが当初の出発点であった。

しかし，デザインに利用することになると，階層ごとの枝の角度，先端の葉の色と形の変化などに細かな制御が必要となり，

表1 リスト

```
%*****************************************************
%  poster.dat  METAMORPHOSE シリーズ 2
%*****************************************************
eye 0 200 -350
view 0 0 1000
angle 39
back 0.6 0.5 0.15
rez 1536 2048
light 5000 5000 -5000 1.0
anti

%*****************************************************
ita = plane 100000 100000        %背景が見切れない様に
rotate ita x -27                 大きな板前に傾けて敷く。
move ita 0 -42 0
mark ita                         %計算時間節約のため
                                 空間分割の例外を宣言。
%*****************************************************
tama_1 = ellipsoid 26 42 26
tmap tama_1 a:¥pic¥gamen1.map  2 1
move tama_1 -80 0 0

%*****************************************************
tama_2 = ellipsoid 37 42 37
tmap tama_2 a:¥pic¥gamen1.map  2 1
move tama_2 0 0 37
move tama_2 -30 20 10

%*****************************************************
tama_3 = ellipsoid 70 70 100
tmap tama_3 a:¥pic¥gamen1.map  6 3
```

```
cut3_1 = plane 300 300
rotate cut3_1 x -30
tama_3 = tama_3 - cut3_1;                    %球の下部切断

cut3_2 = plane 300 300
rotate cut3_2 x 210
tama_3 = tama_3 - cut3_2;                    %球の上部切断

cut3_3 = plane 300 300
rotate cut3_3 z -90
rotate cut3_3 y 30
tama_3 = tama_3 - cut3_3;                    %球の右部切断

cut3_4 = plane 300 300
rotate cut3_4 z 90
rotate cut3_4 y -30
tama_3 = tama_3 - cut3_4;                    %球の左部切断

cut3_5 = ellipsoid 60 60 90

tama_3 = tama_3 - cut3_5;                    %球の内部切断
move tama_3 0 0 100
move tama_3 20 70 40

%*****************************************************
tama_4 = sphere 100
tmap tama_4 a:¥pic¥gamen2.map  6 3
cut4_1 = plane 300 300
rotate cut4_1 x -30
tama_4 = tama_4 - cut4_1;

cut4_2 = plane 300 300
rotate cut4_2 x 210
tama_4 = tama_4 - cut4_2;

cut4_3 = plane 300 300
rotate cut4_3 z -90
rotate cut4_3 y 30
tama_4 = tama_4 - cut4_3;

cut4_4 = plane 300 300
rotate cut4_4 z 90
rotate cut4_4 y -30
tama_4 = tama_4 - cut4_4;

cut4_5 = ellipsoid 90 90 90
tama_4 = tama_4 - cut4_5;

move tama_4 0 0 100

rotate tama_4 z 5

move tama_4 50 120 85

%*****************************************************
tama_5 = sphere 286
tmap tama_5 a:¥pic¥gamen3.map  14 9

cut5_1 = plane 1000 1000
rotate cut5_1 x -10
tama_5 = tama_5 -  cut5_1;

cut5_2 = plane 1000 1000
rotate cut5_2 x 190
tama_5 = tama_5 - cut5_2;

cut5_3 = plane 1000 1000
rotate cut5_3 z -90
rotate cut5_3 y 12.56

tama_5 = tama_5 - cut5_3;

cut5_4 = plane 1000 1000
rotate cut5_4 z 90
rotate cut5_4 y -12.56
tama_5 = tama_5 -  cut5_4;

cut5_5 = ellipsoid 276 276 276
tama_5 = tama_5 - cut5_5;

move tama_5 0 0 286

rotate tama_5 z 10

move tama_5 90 180 130

%*****************************************************
tama_6 = sphere 490
tmap tama_6 a:¥pic¥gamen1.map  18 11

cut6_1 = plane 1000 1000
rotate cut6_1 x -8.1818
tama_6 = tama_6 -  cut6_1;

cut6_2 = plane 1000 1000
rotate cut6_2 x 188.1818
tama_6 = tama_6 -  cut6_2;
```

```
cut6_3 = plane 1000 1000
rotate cut6_3 z -90
rotate cut6_3 y 10
tama_6 = tama_6 -  cut6_3;

cut6_4 = plane 1000 1000
rotate cut6_4 z 90
rotate cut6_4 y -10
tama_6 = tama_6 -  cut6_4;

cut6_5 = ellipsoid 480 480 480
tama_6 = tama_6 - cut6_5;

move tama_6 0 0 490

rotate tama_6 z 20

move tama_6 50 270 190

%*****************************************************
tama_7 = ellipsoid 800 800 800
tmap tama_7 a:¥pic¥gamen1.map  22 15

cut7_1 = plane 5000 5000
rotate cut7_1 x -6
tama_7 = tama_7 - cut7_1;

cut7_2 = plane 5000 5000
rotate cut7_2 x 186
tama_7 = tama_7 - cut7_2;

cut7_3 = plane 5000 5000
rotate cut7_3 z -90
rotate cut7_3 y 8.18
tama_7 = tama_7 - cut7_3;

cut7_4 = plane 5000 5000
rotate cut7_4 z 90
rotate cut7_4 y -8.18
tama_7 = tama_7 - cut7_4;

cut7_5 = ellipsoid 790 790 790
tama_7 = tama_7 - cut7_5;

nuki1 = block  30 30 1605
shin1 = cylinder 15 1610 15
rotate shin1 x -90
move shin1 15 -15 0
nuki1 = nuki1 - shin1;
rotate nuki1 x 6
rotate nuki1 y 8.18

tama_7 = tama_7 - nuki1;    %画面の左上をまるくする。

nuki2 = block  30 30 1605
shin2 = cylinder 15 1610 15
rotate shin2 x -90
move shin2 -15 -15 0
nuki2 = nuki2 - shin2;
rotate nuki2 x 6
rotate nuki2 y -8.18

tama_7 = tama_7 - nuki2;    %画面の右上をまるくする。

nuki3 = block  30 30 1605
shin3 = cylinder 15 1610 15
rotate shin3 x -90
move shin3 15 15 0
nuki3 = nuki3 - shin3;
rotate nuki3 x -6
rotate nuki3 y 8.18

tama_7 = tama_7 - nuki3;    %画面の右下をまるくする。

nuki4 = block  30 30 1605
shin4 = cylinder 15 1610 15
rotate shin4 x -90
move shin4 -15 15 0
nuki4 = nuki4 - shin4;
rotate nuki4 x -6
rotate nuki4 y -8.18

tama_7 = tama_7 - nuki4;    %画面の左下をまるくする。

move tama_7 0 0 790

rotate tama_7 z 25
rotate tama_7 y 15

move tama_7 -100 320 100

%*****************************************************
```

これは、ＲＡＹ－ＴＲＥＫ２による制作を前提にしていますが
誌面の関係で、質感と色の定義は割愛しました。

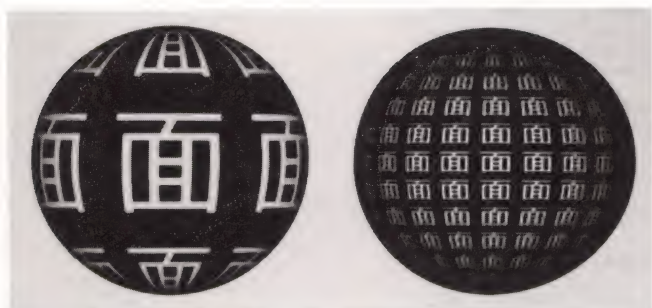
◉図 6 は COLOR IMAGES 参照。

マッピング指定 6：3　　マッピング指定 22：15

図 3

再帰関数では無理があることがわかった。

そこで，枝を1本1本つないでいく手法を見つけなければいけなかった。最初は手探りで枝（円柱）をつないでいったのだが，何度かやっているうちに RAY-TREKⅡの“+”コマンドに着目し，一定の規則性を見つけた。この規則性を C のソースに落として多階層の木構造のデータをつくった（**図 5**）。

最近になって RAY-TREKⅡに “COPY” コマンドが付き，さらに簡単に木構造が作れるようになり，以前 3,000 ラインかかっていたデータの打込みが 100 ライン程度ですむようになった。そのうえ，今までデータを打ち込んでからレイ・トレーシングして形状確認していたものを2次元モデラーをつくることにより，枝ぶりなどはレイ・トレーシングする前にわかるようになった。

図4の作品は，木構造を作ったうえに地面に向かって4灯のスポットライトが当てられ，右上から 1/10 の光源力の点光源が付いている。林檎の実の色は，赤く熟した色から青い林檎の色まで乱数によって定義している。

●林檎の実の色の定義

```
RED   =rand( )/128 * 0.22+199
GREEN =rand( )/128 * 0.7+51
BLUE  =rand( )/128 * 0.1
```

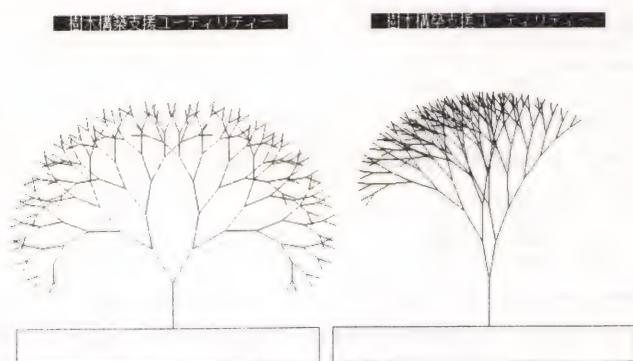



図 5　2次元エディタ出力例

永遠に流れる時間（図 6）

時計シリーズの一つで，このシリーズの他の一つがパンフレットの見開きに使われた。この作品は，それを発展させたもので，“永遠に流れる時間” のイメージを視覚化している。

全体を黄系統にしたのは，20 代前半に筆者が世界中を放浪していたころのエジプト文明のイメージが根底にある。文字盤の中心部から湧き出ている波は，営々と繰り返す歴史の流れを象徴している。数々の波を生みながら流れ続けてきた悠久数千年の歴史は，人々の営みとともに未来へと流れ続ける。その悠大なるイメージを表現したい気持ちにかられるのは私一人であろうか…

ここでは制作過程とともに，基本的な制作手順について説明する。

(1)　全体のラフスケッチをもとに，パーソナルコンピュータでどこまでできるかを物体の面の数などを中心に考える

レイ・トレーシングというと，物体の数が増えると急激に計算時間が増えると思いがちであるが，RAY-TREKⅡの場合，450 サーフィスぐらいまでは実用に耐える時間で計算できる。図4は，ライト1つで質感デフォルトで計算した場合，1時間ほどでできる。そのうえに反射であるとか，透明感であるとかを加えたとしても，計算効率の良いモデリングと無駄のない制作手順を実行することにより，パーソナルコンピュータでも十分計算できる。

この作品では文字盤の数字に時間がかかっている。日常何の気なしに見ている物の構造が，制作してみるとかなり複雑であることがわかる。

図6の作品の文字はすべて角を 45°で切断してあり，その反射が一つの効果になっている。

(2)　文字のモデリング

まずしなければいけないのは，レタリングの資料を集めることである。いくらレイ・トレーシングが美しくてもデザインの基本ができていないものは，質感が美しいぶんだけ逆効果がでてしまう。レタリングの勉強をしたことのない人は，トレースコープで正確に写し取るしかない。ここで重要なのは，できるだけ座標指定が楽になるように作図することである。全体のバランスを崩さないように，なおかつシンプルな数値に置き換えるようにする。

具体的には，1 mm 方眼のグラフ用紙に3面図を起こすときできるだけ 10 mm 単位で作図することである。どうしてもバランスのとれないときは 5 mm 単位にし，最終的にバランスのとれない部分だけ 1 mm 単位を使うようにする。部分をできるだけシンプルにすることが複雑な作品を作る第一歩である。なんだか一般社会の構造のようでもある。

(3)　針のモデリング

一見，簡単そうに見える針であるが，これをモデリングするのには sin, cos, tan, asin, atan, acos など，日常生活ではあまり使わない関数を使わなくてはいけない。しかし，この煩わしい計算も慣れてしまうと “+” とか “−” と同じように使えるものである。モデリングの作業は測量技師の仕事

に似ている。

これらの針は，時間を表すために軌跡を透明にしている。軌跡の質感のパラメータは，数字を連続的にすればよいというものではなく，かなり感覚的に数値を変えてやらなければいけない。数字のうえで連続的であっても視覚的に連続的であるとはかぎらないのがCGの面白いところで，そのノウハウが作品の優劣につながっているようである。図1の黒い卵が黄金色に見えるというのも一種の経験値である。ambi, diff, trans, refl, shine, refr, colorの微妙なバランスから作品が生まれるものである。

過去の画家たちが微妙な色の変化をパレットの中で作りキャンバスに描いていたように，現代の画家（CGデザイナー）はテキストファイルで色を作り，RGBモニターに結果を表示させるのである。

(4) 文字盤について

ある種の幻想的なイメージを創り出すとき，あまりに黒々とした影が落ちていたのでは現実世界に呼び戻されてしまう。そこで存在感と透明感の微妙なバランスが必要になってくる。硬質で，透明で，体温を感じる質感が要求されるのである。

(5) 地面について

地面の質感で最も影響されるのはshine（輝度）である。この調整が全体のイメージを左右するといっても過言ではない。また，透明であり，反射があり，影が落ちているという3つの条件を満たす質感パラメータはきわめて狭い範囲に限られる。これを見つけ出すこともCGデザイナーの仕事の一つである。そのため一度見つけたパラメータは一つの財産であり，他の作品にも流用されることが多い。

(6) 全体について

質感を決定するときは，形を決定するときと同じく迷ったら基本に戻ることが大切である。具体的には，自然界の状態を観察してその現象を再現してやるのである。CGで作画するときは，作りたいものと同じ質感をもった物体を見ながら作画することが大切である。人間は自然の一部なので，自然から学ぶことなしにクリエーティブであることは考えられない。

最終段階として全体の構図の微調整をする。光源の位置，地面の角度，物体の色のバランスなど，全体像が出てからでなくてはできない微調整をするのである。これが終わった後，印刷に適した解像度で最終レンダリングを行う。

C言語とデータ入力

物体を定義する場合，もしくは法線マッピングのデータを作る場合，ことごとく必要となってくるのがC言語の知識である。一定の規則性をもったものや乱数によって表現できるものを一つ一つテキストファイルに落としていったのでは，時間的な問題と入力ミスの問題が重なって無駄な時間を使う場合が多い。このようなときにCの基本的な知識があれば，きわめてスピーディに仕事を進めることができる。もともとそれがコンピュータの基本的な役割なのである。高性能のコンピュータを前にして電卓で数値を求めるなど，滑稽以外の何物でもない。

現在，全くC言語に縁のない人は多少無理をしてでも，社会人向けの講座に参加するなり，身近な人でC言語の知識のある人を探すなりして，習得した方がよいように思う。

筆者自身，まだ最も基本的な使い方しかできないが，CGのデータ入力作業の省力化は目をみはるものがある。人間は絶対に間違いを犯すが，それを克服できるのも人間である。その経験をCのプログラムにしておけば，再び同じミスをしなくてすむのである。

最近CGのモデラーが数多く発表されているが，モデラーにはモデラーの機能の限界があり，それ以外のことはできない。かえって，その限界が創造力の制限になることさえあり得る。モデラーでしかモデリングできないソフトウエアなど論外である。

人間の創造力そのものをディスプレイの中に入れてしまうことはコンピュータに人間が支配されることにつながるのではないかという筆者の危惧は，早晩笑いの対象になってしまうのであろうか？

ヨーロッパにおいて，ベネチア派以来，一流の画家がすべからく自ら岩を砕き，土を練り，画家自身の独特の色合いを見つけてきたように，自分なりのモデリングの手法を開発することが現代のCGデザイナーに必要だと思う。既成のものをマニュアル通り操作していたのでは既存の他の造形活動に勝てるはずがない。これは自らへの戒めでもある。

C言語は，法線マッピングのデータを作るときにも威力を発揮する。例えば，乱数だけでも壁のさまざまな質感の変化が得られる。この乱数という不規則性は，それに条件を付けて制御することにより，さまざまな質感表現が可能になる。レイ・トレーシングには，それがデータ入力というかたちをとっているがゆえに過去のデザイン技法の方が勝る部分がある。しかし，データの作り方いかんでは計り知れない技法の幅がある。それを全面的に支援するのがC言語である。

当社では，独自に画像処理ソフトを開発しているが，処理速度，資料の収集など，開発環境を考えると，現在のところC言語以外の処理系は考えられない。CGデザイナーにとって，今後，C言語は必須の知識になると思う。

ハードウエアとCG制作

当社は，フレーム・バッファにimage makerを使っている。この価格帯のフレーム・バッファとしては唯一ルックアップ・テーブルとRGBの両方が使用でき，この機能は特に1ピクセル1バイトのデータである法線マッピングのデータをつくる際に便利である。RGBとルックアップ・テーブルを相互に渡し合うことによってさまざまなデータをつくることができる。また，RAY-TREK IIのimage maker版はimage makerのメモリに直接マッピング展開するので，512×512のテクスチャ・マッピングと法線マッピングを使用することができる。

（95ページへ続く。）

特集1　グラフィック・デザインにおけるCG

DAI・MEDIA

コンピュータを道具として自由に使いこなし，感覚的に操作できる。デザイン，製版の流れまで変わってしまう。これはデザイン革命である。

杉山　久志*

DAI・MEDIA

DAI・MEDIAとはDigital Artist Interface Mediaの略称で，グラフィック・ペイントボックス，デジタル（ビデオ）ペイントボックス，ミラージュをメインとして，クリエータ自身が創作活動をし，最終原稿までを作り上げるシステムである（**図1**）。

(1)　GRAPHIC PAINTBOX

グラフィック・ペイントボックス（Graphic Paintbox）とは，印刷分野への応用を目的としてデジタル・ペイントボックスの技術をさらに発展させた機種で，走査線2,000本以上に相当するデータを処理できる。従来のコンピュータ・グラフィックス（CG）の概念を変えたばかりでなく，デザイン，印刷の考え方，方法さえも変える可能性をもつ「絵を描くための道具」である。スタイラスペン（styluspen）とタブレット（tablet）を使い，ハイビジョン・モニター上に絵を描く。キーボード（keyboard），ジョイスティック（joy stick）もあるが，特殊用途以外では使わない。操作はビデオ・ペイントボックスと同じで，誰でもすぐに絵を描くことができる。

さらに，レイアウト・スキャナ的機能も合わせ持ち，レイアウト・スキャナ（layout scanner）でできる特殊処理はほとんどすべて可能である。スタイラスペン1本を使い，ハイビジョン・モニター上で合成や特殊処理ができる，きわめて簡単な操作である。従来のレイアウト・スキャナがオペレータがいないとクリエータは何もできなかったのに対し，クリエータ自身が感覚的に操作できることが最大の特徴である。

スキャナで入力し，出力はプルーファー（proofer），MT（magnetic tape），ポジフィルムで行う。また，ハイビジョン映像の入出力も備えている。

図1　左からミラージュ，ビデオ・ペイントボックス，グラフィック・ペイントボックス

(2)　DIGITAL PAINTBOX

デジタル・ペイントボックス（Digital Paintbox）は映像用に開発されたコンピュータである。絵筆やパレットに取って代わりスタイラスペンとタブレットによって絵を描くことができ，スタイラスペンに圧力を加えると色は濃くなり力を弱めると薄くなる。

アーチストはモニターを見ながら自分の描いている絵をチェックできる。油絵，水彩，鉛筆，チョーク，エアブラシおよび

*すぎやま　ひさし　　東洋美術学校　第一デザイン研究室　〒162 東京都新宿区富久町 2-6

通常のグラフィックが可能で，完成した作品は機械的な感じがなくアーチスト自身の作風が表現できる。映像はビデオカメラから入力する。実写や複写を行って，映像の中に手書きで絵を描くことも簡単に行える。また，スーパ機能やアニメーション機能もある。

(3) MIRAGE

ミラージュ（Mirage）は3次元特殊効果装置で，フローティング・ビュー・ポイント・コントロール（floating view point control）により，ペイント・ボックスで制作した絵やビデオの画像などを，難しいプログラミングを必要とせず簡単な操作で瞬時に3次元にできる。三角錐や立方体，球体などの基本形(60数種類）をもっていて，ビデオ・ペイントボックスの絵やビデオ映像をマッピングする。コントゥーア（contour）によってオリジナルの3次元形態を自由に制作することができる。ミラージュは画像の表と裏2画面表示ができ，トランスペアレンス（transparence）も目で確認しながら簡単に行える。モザイク（mosaic），パースペクティブ（perspective）やページめくりなどの特殊効果を利用し，静止画やビデオアートなどを制作している。また，スターライト(starlight)によりライティングができ，光の強さ，広がり，光点の移動などがフローティング・ビュー・ポイント・コントロールにより容易に行える。光の強さの変化により質感もだせる。難しいプログラミングを必要としないので，グラフィックのクリエータには大変便利である。

東洋美術学校では，以上の3機種（**図2**）のコンピュータを中心にして，グラフィック・デザイン，イラスト，写真，レンダリング，パース，版画などの作品作りへの応用研究を行っている。

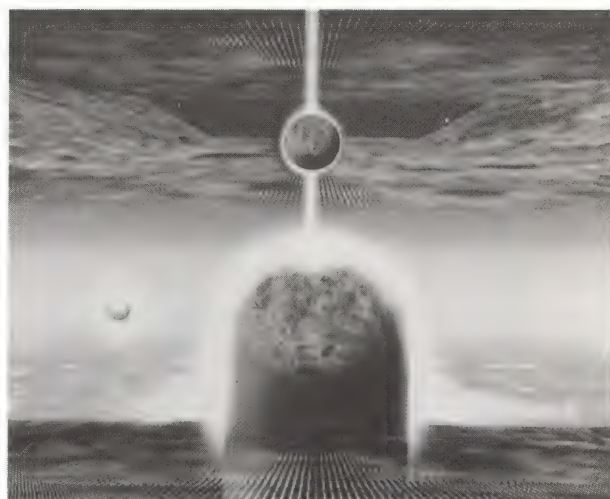



図3　作品例(COLOR IMAGES も参照)

ビデオ・ペイントボックスとミラージュは直結されデータ交換を瞬時に行い，3次元画面でマッピング(mapping)，レイ・トレーシング（ray traceing）を行い，画像を2次元のビデオ・ペイントボックスに取り込み，クリエーティブな作品作りを行っている。また印刷分野の研究もしていて，クリエーティブな作品作りに応用している（**図3**）。

(4) コンピュータ体験

VHS β U-MATIC
1"VTR
VIDEO EDITING SYSTEM
AUDIO MIXER
MIC CASSETE TAPE DECK
COLOUR VIDEO CAMERA
COLOUR MONITOR
COLOUR MONITOR
3.5"FLOPPY
MIRAGE DVM 8020 (3D)
DIGITAL PAINTBOX DPB 7000
HARD DISK (LIBRARY)
HARD COPY CAMERA 4×5"35mm
CONTOUR
FLOATING VIEWPOINT CONTROL
CONTROL STATION
8"FLOPPY
3.5" FLOPPY
HDVS COLOUR MONITOR 28"
CCD SCANNER
HDVS CAMERA
GRAPHIC PAINTBOX
HARD DISK (LIBRARY)
COLOR PROOFER(A3)
MAGNETAC TAPE CROSFIELD HELL SCITEX DAINIPPON SCREEN
CONTROL STATION
HARD DISK (LIBRARY)
DIGITAL FILM RECORDER 8×10"4×5"35mm (DUNN DER8000)
LAYOUT SCANNER

図2 DAI・MEDIA 構成図

コンピュータを使用して作品作りをする場合，作業の途中で自分の頭では考えられなかった画面がモニターに突如現れる場合がある。これは操作上のミスの場合もあるが，本人のミスであってもそのモニターに出た内容が良ければ，その画面から良いところを拾い出し，展開して仕上げることができる。仕上がったものは予定と違ってくる場合があるが，作業の途中に偶然現れた画面であっても，デザインの基礎ができている人はその場で良いか悪いか判断し，良いものだけを正確に拾い出す。そのようにして仕上がった作品には，今まで自分で考えたことのない新しいイメージがその中に表現されている。コンピュータを使用したことにより感性が磨かれ，イメージやアイデアがこれまで以上に広がってくる。

コンピュータで体験したことは，次回からコンピュータを使用しない，これまで通り紙の上で行っていた作業でも，そのイメージを作り出すことができる。この体験をするためには，コンピュータがリアルタイム（real time）に動くことが必要である。一つの命令を数十分，数時間と待つようではクリエータには合わない。作業の途中で熱も冷めてしまう。リアルタイムは第一条件である。

スーパー・デザイニング展

東洋美術学校主催で1988年7月19日から31日まで，東京セントラル美術館においてDAI•MEDIAを使用した展覧会を開催した。浅葉克己，上條喬久，河北秀也，ペーター佐藤，山口はるみなど著名なアート・ディレクター，デザイナー，イラストレータ，フォトグラファーなど28人が挑戦した企画展「スーパー・デザイニング展／デザインが変わる，印刷が変わる」である。タイトル通り，従来のデザイン，印刷，製版の概念を一新する作品を多数展示した（**図4**）。

図4 会場風景

参加したクリエータらは初めてコンピュータを見たときの驚きを隠せなかった。ある人は「革命の真っただ中にいるようだ。これで机の中の定規，烏口，絵の具，紙すべてがいらなくなる」とも言っていた。

グラフィック・ペイントボックスで制作した作品，カタログなどは，データをMTで出力し印刷を行った。作品はB3サイズで出力し，B0サイズの校正刷りで展示した。カタログのページレイアウトもグラフィック・ペイントボックスで行い，作品は原寸でレイアウトし，文字原稿は版下で入稿し，製版作業の時間が大幅に短縮できた。

スーパー・デザイニング展以後，参加したデザイナー，イラストレータ，フォトグラファーらは，ポスター，表紙，雑誌広告，カレンダーなどにグラフィック・ペイントボックスを使用したクリエーティブな作品作りを行っている。

オペレータはクリエータ

グラフィック・ペイントボックス，ビデオ・ペイントボックス，ミラージュを操作するオペレータは，今までのトータルスキャナと同じ感覚で扱うとダメである。キーボード入力のCG作家とも違い，全く新しいタイプのオペレータが必要となってくる。これらのコンピュータは作家とオペレータが一緒に作業を進める。作業の途中で変更がある場合は，オペレータは作家と一緒に考え，コンピュータがもっている機能（オペレータ自身の能力によるところも大である）を使い，適切なアドバイスをしなければならない。そのアドバイスが良くないと，コンピュータがいくらリアルタイムに動いても作業時間がかなりかかり，最終決定が遅れてクリエーティブな作品もできなくなってしまう。

デザイナーがカメラマン，イラストレータに作品を依頼する場合，作品の仕上りはイメージ以上のものを期待している。カメラマン，イラストレータの能力によりイメージ以上のものが上がれば，次回も一緒に仕事をする。このコンピュータのオペレータも同じようにクリエーティブな感覚をもっていなければならない。

グラフィック・デザインへの利用

(1) アートディレクター，デザイナー

ラフスケッチからカンプの制作までの，特にバリエーションを多く作り出すのに大変便利である。現に英国の大手印刷会社BPCCでパッケージのカンプ作りをしたとき，今までの1/3の時間で終わったと言っていた。仕上がったそのカンプも4分待てば，A4サイズ（最大はA3）のプリントが出せる。

今まで指定で行っていた写真，イラストが現物を使用し最終

原稿が制作できる。グリッド(grid)の機能を使い，トンボを作り，実寸を入力してレイアウト，トリミングを行う。印刷する写真やイラストを使用するので，作品はよりクリエータのイメージに近く仕上がり，校正を待って初めて結果がわかるようなことなく結果がすぐに確認できる。作品が出来上がった場合，スポンサーにモニターで確認してもらった方がその場で直しもでき時間の節約にもなる。色校正での変更やミスによる直しもなく，完成度の高い作品ができる。最大のメリットは最終原稿を自分自身でオペレートできることである。

(2) イラストレータ

コンピュータで使用する色はパレット機能 (pallet function) で自由に混色するので，絵の具で混色するのに慣れたイラストレータには使いやすい。ペイント機能中のエアブラシはとてもリアルで細かく書ける。マスクも自由に制作できライブラリにセーブできるので，エアブラシで作品作りをしているイラストレータには大変便利な道具である。スタイラスペンは筆圧で濃淡が決まり，その圧力はエアブラシのガンを引く力と筆圧が同じ感覚なので初めてでも違和感はない。紙に絵の具で描いた場合，描いた後での追加や削除はできない。しかし，カットして大きさを変え，合成して描くということもコンピュータでは自由にできる。また1ピクセル (pixel：picture element) 単位で訂正できるので，細かい作業も容易である。これまでは，紙に書いた作品の大幅な直しは1から書き直すしかなかったが，このコンピュータの場合は写真の修正だけでなくイラストの作品にも十分威力を発揮する。

(3) フォトグラファー

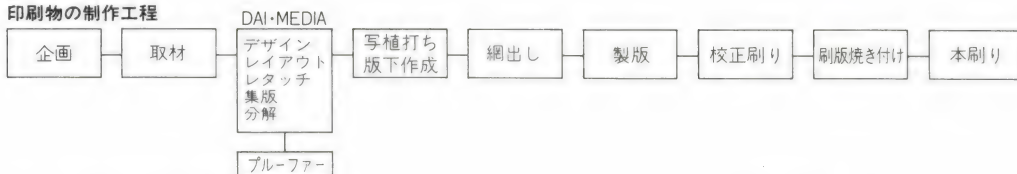
カラー写真のコントラスト，明るさのアップ・ダウンがカラーカーブ (color curves) により簡単にできる。部分的な調子はモニターを見ながら手書きで作り出すことができる。白黒の紙焼きでいう覆焼きの作業がコンピュータによりいとも簡単にできる。ソラリゼーション(solarization)，ネガ(negative)，トーンセパレーション (tone separation)，トーンライン (tone line) など，短時間にいろいろなバリエーションを作り出せ，今まで何重露光していた時間もなくなり，希望の色や形が短時間に作れる。特殊撮影などは分けて撮影し，後でゆっくり合成すれば，時間の短縮と微妙なレイアウトができ完成度がより高くなる場合もある。

カンプから印刷まで

今までのデザインでは，カンプを手書きやコピーなどで作ってスポンサーにOKをとり，印刷原稿は版下で入稿する。写真の合成や訂正，切抜き，特殊製版などはトータルスキャナ(total scanner) を作って行っていた。

デザイナーは校正が上がってくるまで結果を見ることができず，色校正が出てからもう一度考えて校正の段階で大幅な直しを入れることがあった。スポンサーも同じである。これは，デザイナー，スポンサーが仕上りのイメージをしっかりつかみ取れていないからである。トータルスキャナの場合，時間と費用の関係で大幅な直しはやめてしまう。これではそれまでかけた費用はムダになってしまい，良い作品はできなくなってしまう。トータルスキャナで作品を作る場合の一番の問題は，時間がかかり過ぎて，デザイナー自身が作業に立会いチェックする時間が容易にとれないことである。

グラフィック・ペイントボックスは，カンプからフィニッシュまでをモニター上でクリエータ自身の手で行う。一連の流れをコンピュータでリアルタイムに行うので，時間の短縮と完成度の高い作品ができる。

まずカンプであるが，よりリアルな完成に近い形が短時間ででき，バリエーションも多く出せる。コンペなどには強い味方になるはずである。

DAI・MEDIA を使った場合

クリエータは企画の段階から、2次元表現のペイントボックス、3次元表現のミラージュなどを利用して多くのイメージを表現することができ、そのスピードには目をみはるものがある。そのため非常に早い時期にイメージスケッチからカンプまで提出することが可能。企画などの決定されたものはグラフィック・ペイントボックスを利用し、クリエータ自身が短時間で最終原稿まで完成することができる。

完全原稿：グラフィック・ペイントボックスのモニター上で直接制作できる(文字を除く)。

色校正、プルーフ：デジタル・カラープリンタで高性能カラープルーフが得られる。この段階で色校正をすませることもできる。OKならMTに落として印刷所に送り、印刷の仕上がりを待つ。

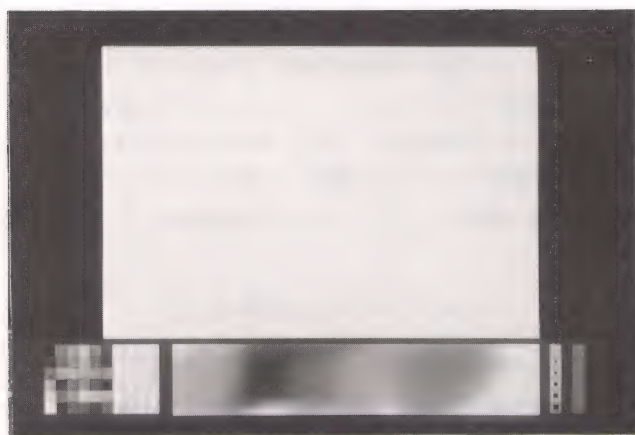
色分解：グラフィック・ペイントボックス内ですませてしまうので印刷所では色分解の必要はない。

MTのフォーマット：今代表的に使用されているCROSFIELD DN SCREEN、HELL、SCITEXに対応する。

MT以外の出力：高解像度デジタル・フィルムレコーダでポジ出力し、従来の印刷にのせることもできる

図5 製版・印刷工程

図6 パレット

図7 ペインティング

ここで気をつけなければならないのが，スポンサーはカンプのプリントアウトを見てOKを出しても，プリンタの問題でプリントアウトによっては印刷で出ない色が出ている場合がある。校正が上がったときスポンサーはプリントアウトの色が良かったという場合もでてくるので，プリンタの種類には十分注意する必要がある。

フィニッシュの写真やイラストは今まで製版所でやっていたが，その作業をクリエータ自身が行うことになる。修正や合成を自分自身の手で行うため，製版所での時間が不必要になる。キャンペーンやシリーズ広告，パッケージなどは特にコンピュータを使用すれば大幅な時間の短縮になる。デザイナーは，そのぶんクリエーティブな時間を多くもつことができる。

写真，イラストなど現物を使った特殊製版のクリエーティブな部分，これはデザイナーにとって大変興味があり最もやりたかったことである（**図5**，前ページ）。

グラフィック・ペイントボックスの機能

印刷に使用するグラフィック・ペイントボックスの機能を中心に主なものを説明する。

(1) パレット（pallet）

パレットに表示されている色は36色である。それ以外の色を作るには，混色，キーボードで数値入力，画面上の色を拾う，の3つの方法がある（**図6**）。

(2) ペインティング（PAINTING）

この中にはペイント，チョーク，エアブラシの3種類のタッチがあり，それぞれ筆圧により濃淡が出せる。エアブラシはパーセントの指定で，より微妙な調子の表現ができる。パレットで混色した色を取り出しモニター上に描く。筆の太さは6種類である（**図7**）。

ティント（tint）で画像の色を手描きで変えることができる。例えば，車のカラーを変更する場合に塗装の質感などを変えることなく青から赤へといったように変更したい部分を書くだけで変えてしまう。

シェード（shade），カラー（color）は明度，彩度を手書きにより部分的に上下でき，写真の調子作りなどに大変便利である。これらの機能は，メニューの選択だけで数値入力をせずにモニターを見ながら自分で確かめて行える。シェード，カラーも筆圧で変化するので大変便利である。

マグニファイ（magnify）は5段階に拡大ができる。細かい修正などは1ピクセルまで確認できて簡単に作業ができる。

(3) エフェクト（EFFECTS）

カラーカーブ（color curves）で，RGB，YMCそれぞれ3色のグラフで，映像全体の色彩，コントラスト（contrast），ブライト（bright），ネガなどの色変換をモニターを見ながら対話形式で行う。

色補正（色のかぶりの修正）やハイライト，中間調，シャドー部などを変更したいときに使用する。

ソラリゼーションなども同様に行う。これらの変換の時間は約2秒でできる。これは使用頻度の高い機能である。

モザイクはデジタル画像を利用したグラフィック効果である。タテ，ヨコ自由に数値入力して行う。マスクを利用して行うことも可能である。

オーバレイ（overlay）はマスクを作る機能である。マスクをかけたい箇所，いらない箇所のそれぞれの部分をペンで選び，明度の差を利用して作り出す。修正が必要なところはドロー・ステンシル（draw stencil）で修正を行う。このマスクは大変便利で，カット，イラストレーション，合成には欠かせない大切なものである。マスクとカットの使い方が理解できると，制作時間が大幅に短縮できる。簡単な機能であるが奥の深い機能でもある。

(4) ペイストアップ（PASTEUP）

画像をカットし貼り付ける。切抜き合成や，左右反転，裏表，拡大・縮小，変形，回転が0.1%刻みで行える。

カットした図柄，文字に対してアウトライン（outline），サラウンド（surround），ソリッド（solid），シャドー（shadow），エンボス（emboss）がかけられる。タイプ（type）は本機が英国のメーカーによる製品のため英文字しかもっていない。ペイストアップの機能の中に含まれている文字は，キーボードによ

図8 ペイストアップ

図9 スキャナ

り入力，カットするとペンに付いてくる。MILES 33 は，文字の編集機能である。レイアウトをするときに非常に便利である（図8）。

(5) スキャナ（SCANNER）

画像の入出力を MT で行う場合，CROSFIELD，SCITEX，HELL，DN SCREEN，DDES の各メーカーのフォーマットをもっているので，現在日本にあるほとんどのトータルスキャナに対応する。6,250 bpi の高密度で行う。また，ハイビジョン・カメラによる入力も可能である。

デジタル・カラープルーファー（digital color proofer）により高性能カラープルーフが得られる。

グラフィック・ペイントボックスと，ビデオ・ペイントボックス，ミラージュは，フロッピでデータ交換を行う。

ディファイン・グリッド（define grid）のプリント・グリッドは，サイズを入力してトンボの制作や寸法の確認を行う。サイズが必要な作業のときは大変便利で，ページのレイアウトなどに威力を発揮する（図9）。

(6) ライブラリ（LIBRARY）

ライブラリには，ピクチャー，ステンシル，カットアウト，パレット，テーブル（カラーカーブ，ブラックカーブ，ブルーフカーブ）がそれぞれ登録できる（図10）。

パレットで色を自由に混色したり，それぞれのカーブで変換したデータをテーブルに登録できるのはありがたい。後日，同条件で作業できる。セーブしたそれぞれの情報は，画像で確認（タイトルでも可）できるので目的の絵が探しやすい。

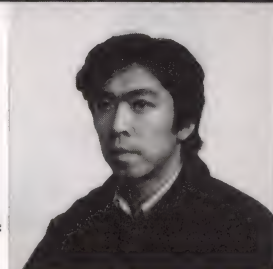
以上の操作は，スタイラスペンを左右に動かすとメニュー，上下に動かすとパレットがモニター上に出る。後は，メニューの選択をするだけの非常に簡単な操作である。

最後にニュースを一つ。英国クォンテル社（Quantel）のバージョンアップしたグラフィック・ペイントボックス XL が今年の秋，日本に上陸する。データは A3 サイズである。少々気になる解像度の問題もこれで解決される。今年の秋が楽しみである。

図10 ライブラリ

グラフィックデザイナーのためのデジタルペインティング入門(1)

ペイントシステムによる作業はデザインワークに何をもたらすのかを，CGを離れた視点から考える。物理的素材から脱したいデザイナーへのメッセージである。

安斎　利洋*

カラーコピー

昨今のデザインワークを刷新したハイテクノロジは何かといえば，残念ながらコンピュータ・グラフィックス(CG)ではなく，カラーコピー*[1]であるという答えが一般的であろうと思う。それはデザイナーの作業ばかりでなく,意識をも改革しつつある。

CGシステムもまた，デザイナーの意識どころか生活態度から晩酌の味や口笛の音色まで改革してしまう道具である，ということに（この場に限って）異論はないだろう。しかし，ここでは一歩譲って，カラーコピーがなぜここまで受け入れられたか，その理由を考えておく必要があると思う。

4つのポイント

カラーコピーがデザイン界を席巻した理由は単純ではないと思われるが，4つの大きなポイントがあるだろう。

カラーコピーの第1の勝因は，対象の形式の斉一性である。カラーコピーは基本的に入口も出口も紙であり，出力を入力に戻すような自己参照さえも可能である。もちろん，ポジフィルムや立体物を対象とすることも可能だが，一度この装置を通過すると他の平面のメディアと均質で平等な，つまり組合せやはめ込みが可能なデータに変換される。

次に，コピーとはいえ，たんに複製を作るのではなく形や色を変換する処理機能をもっている点も重要である。それが貧弱な機能であっても，決して鋏や糊ではできないことが糊付けよりも容易に実現するのである。これは形式の斉一性ともかかわっている。処理を繰り返したり組み合わせたりすることができるから，可能性は無限に広がる。

3番目のポイントは可逆性である。当たり前のことだが，カラーコピーは複製を作る機械である。ある出力は，それに対応する入力と設定されたファンクションさえ整えば再現可能である。つまりやり直しがきく。作業のヒストリが常に保存可能でいつでも過去に立ち返ることができるのは，デザイナーに力強い試行錯誤の勇気を与えるだろう。

最後に，道具としてのメタファーの連続性をあげよう。道具が使い手にその機能を開示するためにはある心的なモデルが必要である。もし使い手に何もイメージさせない道具があったとしたら，それがいかに便利な道具であったとしても受け入れられることはない。カラーコピーはその名の通りカラーのコピーであって，決してスキャナー一体型プリンタでも大きなインスタントカラー写真機でもない。事務用のコピーというきわめて了解しやすいからくりが下地にあればこそ，誰もがその基本操作を自然に想起できる。

さて，本稿の目的は，カラーコピーの優位性を唱えることではない。ここで提示した「形式の斉一性」「イメージの可塑性」「作業の可逆性」「心的モデルの生成」は，カラーコピーに限らず情報を扱うシステムに共通する基本的な問題である。当然，これからわれわれが扱おうとしているテーマであるインタラクティブなCGシステム，とりわけペイントシステムにおいても顕著な問題点であり，特質であると考えられる。

これらをキーワードにペイントシステムは何をしてくれるのか，また何をしようとしているのかについて考えていきたい*[2]。

混在可能な物性

パーソナルコンピュータ（パソコン）によるOA化が著しく進み，オフィスからペーパーが消える，という話は何年も前から聞かされている。にもかかわらず，なかなかカミの恩恵から自由になることができないのは周知の通りである。

推察するに，これは次のようなことに起因するのではないだろうか。

パソコンの吐き出す情報をプリントアウトすれば新聞の切り抜きなどと等質の形式になる。しかし，反対に新聞の切り抜きは，人手で一文字ずつパソコンに入力するほかない。われわれを取り巻く情報はすべて文字コード化されているわけではないから，引用できる情報とできない情報ができてしまい，対象が均質にならない。テキストデータとイメージデータを組み合わせても，それを料理することはできないのである。

当たり前のこととして承服しがちだが，これは重要な欠落であると思う。同じ舞台に立っていなければ相互に参照することはない。鏡を向かい合わせると空間が広がるように，相互参照は足し算以上のものをもたらすだろう。情報を組織化するうえ

*あんざい　としひろ　㈱サピエンス　CGソフトウェアルーム　〒169　東京都新宿区高田馬場4-9-11

図1　ニブ一覧

図2　セル板

でデータ形式の統一は不可欠なのである*3。

これに対してペイントシステムの素晴らしい性質*4は，対象となる画像はすべて平等であるということに尽きるのではないだろうか。およそ映像と名のつくすべての映像を，現実的な装置によってシステムの中に取り込むことができる。例えば，手書き画像，印刷物，写真，立体，ビデオ画像，CG 画像，これらが等質の素材として同じ舞台の上に並ぶのである。光学的・物理的素材の束縛になれ親しんだデザイナーにとって，これは大きな意識革命である。

例えば，乾いていない油絵具を指で擦ると混ざりながらかすれていく。水彩絵具に水をたらすと不思議なパターンを残して拡散していく。物理的な世界では水と油ほど違うこれらが，ペイントシステムの上では均質かつ混在可能なのである*5。例えば，パステルのように擦りつけるエアブラシや染み込むチョークも即座に作れる*6。

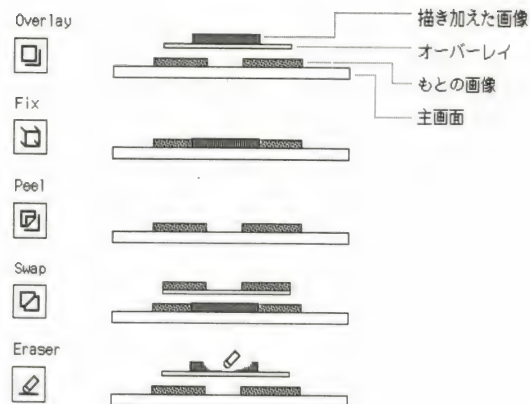
そればかりではない。スキャナから読み込んだ写真もまた指で擦ったり，水をたらしたりする対象になる。また，水で拡散したパターンが，その場で光学的な処理や暗室での特殊効果処理にかけられる。

また，ペイントシステムで作られた画像は3次元のレンダリング・ソフトによって立体の表面に貼り付けられ，その結果は再びペイントシステムの対象になる。

この自在な参照性は，すべてが斉一な画素配列情報であるという性格によるものである。これほどまでに自己増殖の可能性を秘めた造形システムは，これまでどこにもなかったのである*7（**図1**）。

可逆的な紙

ペイントシステムは，たいていやり直しのための画面を備えている。それは現在作業をしている画面をそっくりそのまま保存するもので，新たな作業を打ち消して保存した時点まで画面を戻すことができる。人間は必ず間違いを犯す動物なので，やり直し（アンドゥ）はできるに越したことはない。

さてこのやり直し画面は，たんに消極的な予防策にとどまるわけではない。ペイントシステムが対象とする空間は，基本的にフレームメモリの画素配列である。しかし物理的な素材同様，これは決して2次元空間にとどまっているわけではない。やり直し画面はいわば2次元に時間軸を交差させる。

この画面を利用すると，次のようななじみのテクニックを用いることができる。

- カラー・オーバレイやスクリーントーンを画面に貼り，不要な部分を切り取る。
- 盛り上げたアクリル絵具を削ってその断面を表す。
- パステルに練りゴムをあて，定着液をかけた過去の画面まで遡る。
- 故意に濃く描いた像をうっすら半透明に消していく。

水晶玉とあだ名されたレンズがついているシステムでは，過去の画面がどうなっているのかを現在の画面を壊さずにのぞき見ることもできる。

過去の画面に描き加える，過去に戻すという時間軸上の作業は，ここでは重ねる，剝すといった画面の奥行き方向への作業と等価である*8（**図2**）。

心的モデル

われわれを取り巻く情報や情報処理機能は，なんらかの心理的な像をもたないかぎり人間にとって無に等しい。そこで最も役立つのが，既存のシステムの隠喩を用いるいわば「本歌どり」である。集計用紙や情報カードといった紙の名前がコンピュータのマニュアルを埋めるのはそのためである。

ほとんどの人が，ペイントシステムを起動するとまず真っ先にマウスやスタイラスペンを手で動かしてみて，画面中のカーソルの連動を確認するに違いない。そして，ボタンをクリックしたりペンを押し付けたりして，画面がどう反応するかを観察する。ペイントシステムは，対応する画素およびその近傍に対してなんらかの書換えを施すだろう。

するとわれわれは，ここには「何か」があり，われわれはその「何か」に作用を及ぼす「道具」を握っている，ということを了解する。「何か」の実体はフレームメモリであり，「道具」はプログラムであると考えている人はおそらくいないだろう。普通，これらは紙，絵具，筆というメタファーで説明されるし，説明されなくてもこれがなじみの道具であることに抵抗を感じる人はいない。これは不思議な精神の作用である。ペイントシステムはこのような心理的な像の喚起力によって，まず使い手の前に現れる。

目の前にあるものが紙なのか，あるいは紙のような映像であるのか，それは人間にとってさほど重要な問題ではない。目の

前の紙が，丸める，汚す，破くといった働きかけに対してどのような反応を返してくるかがその紙に対するわれわれの認識のすべてである。たとえ本物の紙であっても，汚してもしみ一つ残らず丸めても皺一つ残らなければ,もはやそれは紙ではない。反対に、ある働きかけに対して的確な反応さえ返せば，それは幻であっても心理的には紙と等価である。デザイナーは，物理的な束縛を全く受けない文字通り「幻の紙」を手に入れたということができよう。

注意深く設計されたペイントシステムであるなら，「紙」以外にも多くの「何か」を発見できるはずである。これは溶剤，これは消しゴム，これは定着液というように。あるいはまたシステムとの対話の中で，擬人化されたアシスタント*9に出会うかもしれない。こうした仮想の物性や人格を作り上げるために，ペイントシステムは整合性のあるレスポンスを可能なかぎり素早く返さなくてはならない。もし，矛盾した動きがあったり，反応が滞ったりすれば，たちまち幻の「何か」は目の前から消え失せる。ペイントシステムは，あくまでイリュージョンを操るシステムである*10。

今後の課題

道具は人間の身体の延長であるという考え方がある。ナイフは手の，車は足の，カメラは目の，それぞれ延長と考えるわけである。

さて，それに従えば，コンピュータは脳の延長ということになるようである。例えば大量のデータを統計的に扱ったり，そこから意志決定を支援する論理を展開したり，そういったコンピュータの姿は人間にとって外的に進化した脳とよぶにふさわしい。

しかし，やはりコンピュータと密接に結び付いているとはいえ，ことペイントシステムに関するかぎり，このメタファーにはいささか無理がある。少なくとも，筆やエアブラシをポインティング・デバイスに持ち変えたデザイナーにとって，そこにあるシステムは脳の延長とよぶにはあまりに素朴である。むしろ努めて何気ない手の延長として，また目の延長としての道具であろうとしているように思える。オペレータは相変わらず画像をルーペで拡大しては，頻繁にマウスの頭をこずきながら力づくの作業をしている場合が多い。

われわれのペイントシステムが，ある日美意識に目覚め，デザイナーがボタンをクリックしながら片付けている膨大な判断を肩代りすることもあるかもしれない。このようなコンピュータ・エイディッド・ペインティングが実現すれば，それはまさしく脳の延長とよぶにふさわしいだろう*11。

しかし，たとえ美に目覚めなくても，われわれの脳はごく単純な判断の繰返しに酷使される場合がある。例えば，人間の顔を切り出すとか，画像のエッジにある不自然な切り残しを取り除くといった場合，やはりどうしてもルーペの力仕事に頼らざるを得ない場合がある。この程度の判断，つまりどれが境界線か，またどれが不自然なノイズか，そういう判断の自動化がこれからのペイントシステムの課題であろうと思われる*12。

すでに，いくつかの画像処理機能を組み合わせた，ノイズ除去やエッジ抽出は可能である*13。しかしそれでも，1画素ごとに手で処理する機会から逃れるのは難しい。力づくの作業をいかにエレガントに解決できるかが，コンピュータを使う意義の有無を分けるのである。

〈補足〉

＊1：もちろん，某CG専門誌と同じ名前をもつ製品のことである。

＊2：目下われわれが作るべきなのはペイントシステムではなく，よりインテリジェントなカラーコピーかもしれないと考えている。この新しいコピーは，原稿を読み取ってすぐに紙が出てくるわけではない。画面の上で修正，レイアウトができるし，その画像を記憶しておくこともできる。つまり，これはペイントシステムそのものであるが，カラーコピーという下地をもち，それを置換することによってこれまでとは違った意味の場にCGを組み込むことができる。

＊3：この意味でOCRの普及は重要な意味をもつと考えられる。

＊4：正確には「デジタル画像の素晴らしい性質」というべきである。本稿は，CGの中でのペイントシステムの位相ということにふれるつもりはない。グラフィックデザインにペイントシステムが何をもたらすのかを，デザインワークの流れの側から考えるのが目的である。したがって，ペイントシステムがCGシステム全体を指す場合もあることをお断りしておく。

＊5：筆者らの手によるペイントシステムSuperTableau, SuperTableauPREMIUM,ESUQUISEなどは複合的な物性を自在に組み合わせることができるように，次のような筆先を用意している。

Pen：ペン領域を塗りつぶす。

RandomColor：岩絵具。あるグラデーションからランダムに色を取り，ペン領域を塗りつぶす。

Push：絵具を擦り，混ぜる擦筆。ペン領域の大きさの毛細管の束を想定する。各毛細管は画面から色を取り，再び画面に戻すことを繰り返す。筆を動かすことによって絵具を擦り，混ぜる効果をだせる。

Defocus：レンズでぼかすようなフィルターの効果。各画素について，8近傍の画素の平均の色を重ねる。

Knife：ペン領域の色の平均をペン領域全体に均一に広げる。油絵のペインティングナイフの効果。

Sander：各画素の彩度を上げ，その分だけ近傍の画素の彩度を下げる。繰り返しなぞると，その部分の色調を保ったまま砂目状に画面を荒らす。

Blot Brush：インクを画面ににじませる。ランダムな分岐によって独特のパターンを描くフラクタルペン。成長点の終了条件によって筆の広がりを制御している。先端にいくほど絵具の濃度を下げるので，紙に絵具がにじむ効果になる。

Water Brush：画面に水をたらしたようなぼかしを入れ

る。色彩をもたない筆先がランダムに分岐し，画素の色を移動させる。その結果，画面に水をたらしたようににじんでいく。

Air：いわゆるエアブラシ。周囲にいくほど減少する濃度勾配をもたせている。

Particle：砂目のエアブラシ。ペンの中心からランダムに隔たった画素に絵具の荒い粒子を飛ばす。

Copy Brush：画面を一定のベクターだけ離れた画面からコピーする。

Eraser：消しゴム。過去の画面を透視する。

Smooth Brush：アンチジャギー処理を施した線をねった毛筆。フリーハンドの場合，描画点とカーソルの距離により筆の太さと広がり方向を制御する。

＊6：浸透性のインクかそれともパステルのような擦りつける顔料かは，物理的な画材では大きな違いだが，プログラムのうえではそれほど大きな差はない。一定時間ごとに処理を繰り返せばインクになり，もし同じ画素に続けて処理しようとしたときにそれを禁止するとパステルになる。ほんの少しの差が全く異なる物性として現れてくるというのが，ペイントシステムの大きな特徴である。

＊7：ここで言及しなかった問題がある。それは新聞の切り抜きと同様，文字の問題，そしてベクター情報の問題である。文字と図形データは高度に圧縮された図像であり，それを画素配列に展開すると再圧縮できない。そこで文字や図形は別形式のデータとして管理されることになる。当然そこには「自在な自己参照性」という特質は生じ得ない。

DTP のフルカラー化に伴い，こうした問題はペイントシステムとも無縁ではなくなってくる。

＊8：消しゴムなどの機能は，主画面と同等以上のメモリを必要とする。そういうハードウエア上の制限から，消しゴムが実現できないシステムもある。この水晶玉がついているのは SuperTableauPREMIUM である（もっとも，正式には X レイ機能とよんでいる）。

SuperTableau の場合，セル板とよばれる板の上で半透明の消しゴムが使えるようになっている。セル板は，メモリ上に取られる小さな Undo バッファで，これを仮想の透明板と考えて理解の助けにしている。セル板について次のような操作が用意されている。

Overlay：セル板を画面上に置く。

Fix：セル板を画面に固定する。

Swap：セル板と画面を交換する。

Peel：セル板を剝し，セル上の仕事はキャンセルする。

Eraser：セル板上の作業だけ部分的に消し去る。

＊9：アシスタントの条件はいろいろあるが，判断を仰ぐべきことはしっかり聞き，あまり重要でないことはしつこく質問しないというのが鉄則である。

SuperTableau では，システムからの問いかけに「無視できない質問」と「無視できる質問」という2つの階層を設けている。「無視できない質問」は，人間が判断を下さないかぎり次に進めない。「無視できる質問」は，その場で即決すれば GO を指示できるが，次の動作に移ることでキャンセルできる。

対話型 CG システムの教科書として名高い Fundamentals of Interactive Computer Graphics によれば，インタラクティブ・プログラムを書くにあたっての指針として，①対話の進行が簡単で統一されている，②多くのオプションやスタイルでユーザーに負担をかけない，③対話の各段階でプロンプトを与える，④適切なフィードバックを与える，⑤力づくでなく誤りから回復できる，といった事項をあげている。これはプログラムを書く側への指針であるが，同時にこれらはアシスタントを評価する側のチェック事項でもある。

＊10：ペイントシステムがいかに心理的なシステムであるか，次のような示唆的な実験がある。同様の結果をもたらす A と B の2つのルーチンがあり，A の方が B よりも速いにもかかわらず B の方が A よりも視覚的に速く感じる。フリーハンドで A，B 2つの筆を使い分けると B の方が仕上がりが美しい。メンタルなシステムでは物理的な時間より心理的な時間を優先した方がよい場合がある。

＊11：CG 作家の梅村高氏は，画像各部の心理的な重みをテーブルとして与えることによって，それをもとに特殊効果を付与するシステムを提唱している。美の判断基準は，そう簡単に脳を離れることはないわけである。

＊12：ある特定の条件を満たす画素を発見するために，SuperTableau ではオートマスク機能を用意している。オートマスクはフルカラー画像を参照し，色彩がある条件にマッチした画素に対してマスクをかける機能である。条件によって8つのプロセスがある。

クロマキー：厳密にある一色にマッチする部分だけマスクする。

H：ある特定の色相だけマスクする。色相を指定するヒューテーブルは画面から設定できる。

S：ある彩度より上（下）の部分だけマスクする。

V：ある明度より上（下）の部分だけマスクする。

I：ある明るさ（R+G+B）より上（下）の部分だけマスクする。

R，G，B：赤（R）の明るさがある値より上（下）の部分だけマスクする。G，B も同様。

今後，空間フィルタと組み合わせてまわりの画素との関係も判断材料にするようにしたいと考えている。

＊13：主に科学技術用に用いられる画像処理機能の多くのものは，グラフィックデザインでも有用なものである。その中には，まわりの画素から判断して突発的なノイズと思われる画素を，周囲の代表的な色に置き換えるといった膨大な手作業を瞬時に肩代りしてくれるものもある（ランクバリュー・フィルタ）。

次回からは，具体的な作例をもとにさまざまな作画技法を紹介していく予定である。

パソコンによるプリ・プレス処理

今やパソコンはミニコンの機能を超え，不可能といわれていた印刷製版のプリ・プレス処理が可能となった。実際にシステムを使用しての現場からの報告をする。

行木　修*

現代のコロンブス

今日もまた，すべてのコンピュータが稼働している。スタッフたちは忙しそうにコンピュータと向かい合っている。私のスケジュール表は今日も来客のアポイントでいっぱいである。その内容は，ビデオ・アニメーションの打合せ，印刷用の写真合成の打合せ，システム導入のコンサルティングなどさまざまである。毎日がこんな繰返しで過ぎていく。

当社は，主に新聞用の白黒製版を行っている企業である。他に子会社としてデザイン部門を2社，写植部門を2社グループ化しているが，私たちは本社の中で昭和ビデオセンターというビデオ制作部門の一部としてC-GAM（Computer Graphics Art&Magic）というコンピュータ・グラフィックス(CG)部門を動かしているわけである。今後の「パーソナルコンピュータ(パソコン）におけるプリ・プレス処理」というテーマにふれる前に，まず当社の機能と立場を知っていただきたいと思う。

前述したように，当社の本業はデザインと製版である。ところが，6年ほど前に映像出版という言葉が盛んになり，多くの印刷製版会社がビデオ事業に参入したときに，昭和ビデオセンターもスタートしたのである。しかし，ビデオ事業はすでに競争が激しく，他社との差別化を考えなければならなかった。

そこで，当社で導入したのが日本ビクター製のアニピュータというCGシステムであった。256×220ドット，26万色中256色表示という機能でリアルタイムでアニメーションができたのである。ところが当時は，CGがまだ普及しておらず，当社もメーカーもどのようにしてシステムを応用していってよいかわからず苦労したものであった。このころを第1次CG導入期と私たちはいっているが，デザイン，ビデオ，印刷，ファッション業界を含め，かなりのCGシステムが導入された。しかし，その大半はシステムを完全に理解して動かすことができず，毎月のリース料のみを支払う羽目になってしまったのである。

そんな中で，私たちは子会社からのデザイナーの参加によりCGを制作し始め，CGイラストやCGアニメーションなどの技術もしだいにアップし，CGによるTV-CMアニメーションや雑誌広告，ポスターまで制作するようになっていった。

一方，当初社内の制作のみを行っていたが，ビデオ・プロダクションや印刷デザインなどの仲間業者からの制作依頼が増え，システムのグレードアップも考えなければならず，640×480ドット，1,677万色表示のimage makerを導入することになった。image makerでは2次元と3次元ソフトを組み合わせ，イラストのみならずパッケージデザインやID，建築シミュレーション，環境デザインなどを行い，本格的なCGL（Computer Graphic Laboratory）の機能とシステムを販売するディーラー機能，CGの教育をするセミナー機能を含めCGに関するすべてのサービスを行うこととなった。しかし，まだこれらのCGシステムで制作した作品は解像度が低く，フィルム出力をしてもとても写真レベルで使用することはできなかった（俗に640×480ドットくらいの解像度はビデオレートといい，ビデオ一画面の細かさをもっている）。そんなとき，1987年3月にイマプロのCGシステムと出合ったのである（後述）。

*なめき　おさむ　　昭和製版㈱　C-GAM　〒531 大阪市北区大淀南1-2-22

ミニコンの時代からパソコンの時代へ

CGのシステムにもいえることだが，この1～2年のパソコンの進歩には目覚ましいものがある。以前のパソコンというと，価格は安いが計算が遅い，記憶容量が小さいなどの問題点があった。しかし，32ビット機の出現，20MHzのクロック，20MIPSの計算速度，600Mバイトのハードディスクのサポート，あげくは2.2Gバイトの磁気テープ(MT)やMS-DOS Ver.4.0のサポートにより，ユーザーズメモリを無限に使用できるなど，すでに大きくミニコンピュータを超えてしまっている。そして，CPU本体の機能ばかりでなく，周辺機器の充実にも目をみはるものがある。例えば，入力側では600dpiのフルカラー・スキャナの登場，ハイビジョン・ビデオカメラとハイビジョン・フレームバッファの登場など，メモリ側では追記型光ディスクのローコスト化，1GバイトのDAT-MTの登場と，2.2Gバイトの8mmビデオテープに記録するMTユニットなどは今後のメディアを大きく変えるであろう。また，出力側ではレーザー機器のローコスト化により，レーザー・フィルムレコーダ，レーザービーム・プリンタ，レーザービーム・プロッタ，フルカラー表現が可能な熱昇華型フルカラーA3プリンタ，A0インクジェット・フルカラー・プリンタなどのデバイスが登場してきた。このシステムはすべてパソコンに接続可能であり，こうなると，これまで一部屋を占領していた高価なミニコン・システムが，今後パソコンに取って代わられる時代がやって来るのは必至であろう。

今，世界はDTPからプリ・プレスへ

パソコンというと日本のお家芸のように思えるが，やはりアメリカに比べると遅れている部分が多い。それはハードウエアの機能面ではなく，周辺機器やソフトウエアのノウハウ部分に特に多いようである。特に，業務レベルで使用できる周辺機器をつるして，それにソフトウエアを開発し，本格的に使用できるシステムを開発するとなると，やはりコンピュータの歴史の長いアメリカの独壇場になってくる。ゲームマシンとして有名であったアップル社が日本のパソコンに押されて一時危機説があったのを，DTP(Desk Top Publishing)という考え方によりレーザービーム・プリンタをサポートし，版下作成をパソコンで行う画期的なアイデアで世界の市場に復活した例などは，ハードウエアとソフトウエアを巧みに組み合わせた成果である。今，アメリカでは，アップル社のマッキントッシュをはじめとしてIBM PCなどパソコン主流の時代がやって来ている。しかし，そのベースには通信ネットワークがすでに全米，全世界にネットされ，ユーザーはそのネットワークを利用し，2,000dpi以上のレーザープロッタを制御してYMCKの4色分解刷版を作成するプリ・プレスシステムが登場し，しだいに普及し始めた。

クリシュナコピー社は全米に30店のネットを持ち，マッキントッシュで作成した絵や文章をカラープリンタやレーザープロッタで出力し，カラープルーフ（校正刷り）サービスやカラーコピー・サービスをはじめとし，他の支店と通信網によりデータの送受信を行っている。

●図1は次ページ。

また，印刷製版業界には以前からトータルスキャナ・システムといわれるプリ・プレスシステムがあるが，そのシステムと接続し，サブシステムとして使用されている例では，全米同時発売のフルカラー新聞「USA Today」紙があげられる。ここでは，イスラエルのサイテックス社のレスポンスというトータルスキャナを使用しているが，そのシステムとカナダのイマプロ社のプリ・プレスシステムをオンラインで接続し，作業の効率化を図っている。

一方，米国ムーア社でも同じようにレスポンスとハンドシェイクし，デザインシステムとして利用されている。このようにアメリカではさまざまなかたちでパソコンによるプリ・プレスシステムが稼働し始めている。このベースには，パソコンによるグラフィックシステムの普及という大きな要因が必ず考えられる。パソコンで作成してイラストやグラフィックスをなんとかデータから印刷へもっていきたいというユーザーたちの熱意が，このようなシステムを生むきっかけになったのだと思う。

プリ・プレスとは

プリ・プレスとはPre Press Systemのことで，印刷前システムと直訳できる。つまり，印刷の前段階であるカラー分解，刷版作成までをいうのである。基本的には，あるクライアントから仕事が発生すると**図1**のような流れを経て仕事が進む。まず，クライアントとの打合せで，ラフといわれる全体のイメージを大ざっぱに表現したものが作られる。これには，現在カラーや白黒コピーを用い，拡大・縮小した素材を貼り合わせてイメージを表現しているデザイナーが多い。次に，そのラフをクライアントに見せて全体のイメージを詰めていく。この段階では，ラフを書き換えて何度もプレゼンテーションを行うことが多い。ようやくラフが決定すると，次に制作段階へ移る。つまり，実際に文字を写植打ちし，イラストなどは書き上げ，撮影なども行っていく。そして次にカンプへと移る。これは実際の使用原稿の大きさの台紙に写植文字やイラスト，写真などを貼り込み，定成に近い状態でクライアント・チェックを受けることであるが，実際は時間とコストがかかるため，印刷段階の色校正でこのチェックを代行してしまい，あまり行われていない。そしてその先に版下作成がある。版下は実寸の台紙を使用し，文字や罫線のように網をかけない部分と写真などのように網をかける部分などに分け，レイアウトされて色の指示が入った状態のものをいう。

これができると，ようやくデザイン部門の仕事が一段落し，次に印刷段階へと移っていく。これを印刷出稿という。デザインと印刷製版が分かれている企業では，この段階で印刷製版会社に版下が渡される。

さて，渡された原稿は網をかけるもの，または100％の網でよいものに分けられ，網かけの必要な写真やイラストのようなものはカラー分解といって印刷の4色のカラーであるC（シアン），M(マゼンタ)，Y（イエロー），K（墨網）に分け，おのおののフィルムに出力される。その後，文字や写真などへ網を入れたり，入っていないフィルムを切り貼りしたりして1枚のフィ

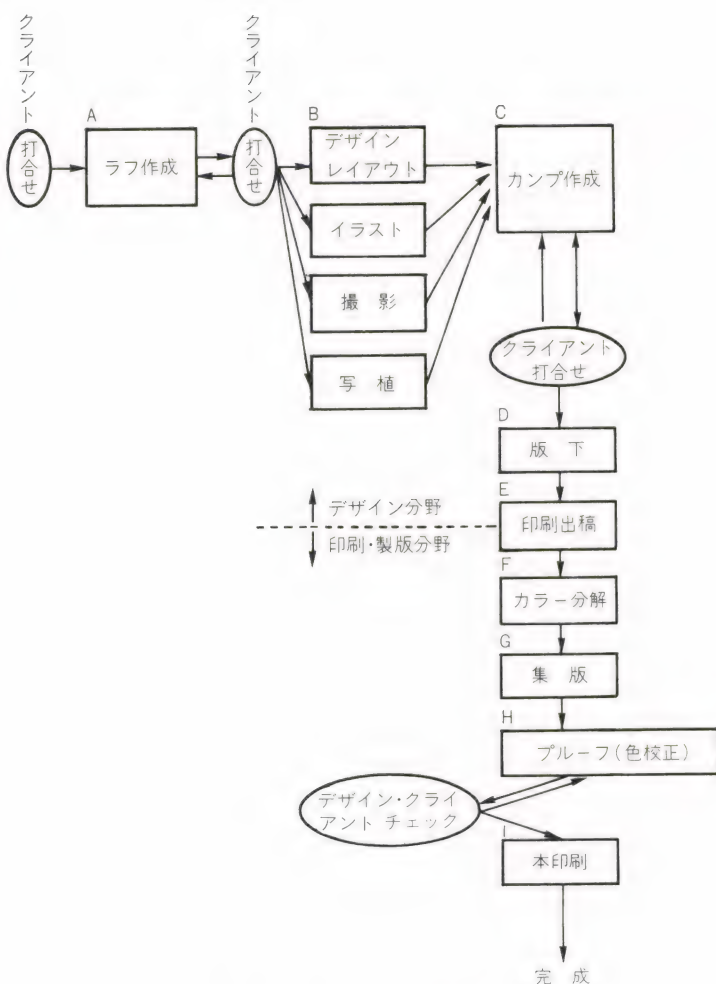



図1 デザインから印刷までの流れ

ルムに貼り入む集版作業があり，この段階で1ページ分の刷版が出来上がるのである。集版作業が終わると色校正があり実際に試し刷りを行い，印刷製版会社ではカラー分解の具合やキズの有無などをチェックし，デザイン側では指示した色のバランスなどをチェックする。

これが終わるとようやく印刷段階に入る。初めて印刷機にPS版が組み込まれ，印刷が始まるのである。デザイン段階を別にしても印刷製版段階で約1カ月弱という，かなり時間のかかる作業が従来行われてきたのである。デザイン段階も仕事の内容で多少異なるが1～3カ月くらいかかることは日常茶飯事である。

プリ・プレスシステムの必要性

プリ・プレスシステムは略してPPSあるいはIPPS (Image Processing Pre Press System＝画像処理印刷前段階システム）といわれているが，印刷製版業界では7年ほど前からトータルスキャナ・システムとして知られている。図1のEからFまでの段階を行うシステムとして，HP-1000クラスをCPUとし，並列処理機能や大容量のハードディスクやオープンMT（現在では特に大容量とはいえず，パソコンのデバイスの方が容量は大きいものがある）を生かし，数千万円もするドラム型スキャナや平面型スキャナを使用し，画像を入力処理し，レーザープロッタなどの出力する電子製版システムである。大手では大日本スクリーン製造のシグマシリーズをはじめとし，サイテックス社のレスポンスシリーズ，英国クロスフィールド社のマグナスキャン，西独ヘル社のクロマコムなどがある。

本来，これらのシステムはコンピュータによる幅広いクリエーティブデザインの可能性と製版段階の効率化を目指すものであり，アメリカでは六百数十台導入されているが，うち約半数以上がクリエーティブ中心に使用されている。これに対して，日本では四百数十台の導入のほとんどが製版段階の集版作業が中心である。これは非常に残念なことであるが，これに関してはさまざまな原因が考えられる。まず第1に，システムが1億円以上と非常に高価で，時間のかかるデザインのために専有させられない。第2に，アメリカではパソコンCGはすでに10年以上の歴史があり，その歴史の上にトータルスキャナ・システムが乗ってきているのに対し，日本ではCGはまだ4～5年の歴史しかなく，そのうえパソコンCGに全く手をつけていないデザイナーや印刷業者がほとんどで，基本的にIPPSによるクリエーティブ作業が理解できていない。以上の2つの原因により，日本でのプリ・プレスシステムの利用例はクリエーティブ性に欠けている。

まだ当社に低解像度やビデオレートのシステムしかなかった当時，作品をフィルム出力し，印刷原稿として使用した際，よく印刷業者やデザイナーから「粗くてドットが見える」と言われたことがある。根本的にはビジュアルの面白さであって，粗かろうが細かかろうがそれは問題ではなく，問題なのは企画であり，ビジュアル・コンセプトであるということがわかる人が残念なことに現在でも少ない。つまり，きれいに印刷することばかりを考えていて，全体のコンセプトをみることができないのである。

今，私たちの最も重要なパートナーの一人として，カナダのイマプロ社のフレッド・アンドレオーネ社長がいる。2年前にイマプロ社から発売されたパソコンによるIPPSは，当時日本ではスライド作成システムとしてしか評価されなかった。なぜなら，解像度4,096×3,072ドットの出力は35mmエクタクロームのエマルジョンとほぼ同等で，35mmフィルムは日本では印刷原稿にあまり使用されず，したがってスライド作成システムだというのである。

ところが，アメリカでは印刷原稿は35mmが大半で，それ以外は8×10になり，日本のように4×5サイズはあまり使用されない。ところが，日本でも著名カメラマンは35mmで撮影した原稿を大版ポスターに使用している。つまり，35mmサイズでも印刷原稿としては十分なものもあるのである。そういう点からみても，おおらかなアメリカはいつもアプリケーション面で日本をリードしている。

パソコンによるプリ・プレス

図1を見ての通り，A～Hまでをコンピュータによる処理で作業を進めることができたとしたら，計り知れない効率化が考

えられる。これに関しては，トータルスキャナ各社も着眼し，パソコンやワークステーションによるデザインシステムで全作業を行うことを考えている。一方，パソコン側からみれば，前述したようにメモリの増設や各種デバイスの開発により，デザインシステムからプリ・プレスシステムへ機能を上げていくことが可能になってきた。これがパソコンによるプリ・プレスシステムの登場である。プリ・プレスシステムを正確にとらえ，デザインからカラー分解，集版，プルーフ，刷版までのトータルシステムとして考えると，現在，パソコンでは，アップル社のマッキントッシュIIを使用したシステムとイマプロ社のリタッチャーシステムしかない。前者は出力機にライノトロニックやバリタイパー，後者はECRMなどのレーザープロッタを接続し，印刷用4色刷版を出力する。主な特徴として，マッキントッシュではDTPの流れから文字の考え方が強く，Page MakerやReady Set Go, Quark Xpressなどのデザインレイアウト・ソフトに日本語を載せているが，フルカラー(1,677万色)表現はできないので，写真をページ・レイアウトしたり写真合成やレタッチはできない。イマプロ社のリタッチャーでは，その名の通り写真のページ・レイアウトや写真合成，レタッチをはじめとしてグラフ作成，CGイラスト，文字などを一括処理できるうえ，サイテックス社のレスポンス・シリーズなどと双方向通信が可能である。

パソコンによるプリ•プレスシステムを使用して

さて，いよいよパソコンのプリ・プレスシステムを使用しての実際の作業を見ていこうと思う。前述のマッキントッシュのシステムよりイマプロ社のリタッチャーシステムの方が，プリ・プレスシステムとしての入出力系および処理内容はより完成されているので，このシステムを使用して作業の実際を紹介しよう。

イマプロ社リタッチャーシステムの全容は**図2**の通りである。入力系は300 dpiスキャナ(主にA3までのプリント原稿からの入力)，600 dpiスキャナ（4×5オリジナル・ポジをダイレクトに読む)，35 mmスキャナ（35 mmスライドをダイレクトに読む)，ビデオカメラまたはVTR（ビデオ画面を入力する)，MS-DOSによるアスキーファイルで記述されたデータなどが使用できる。CPUに入力された情報は写真を合成したり，修正したりするピクセル・レイヤーとグラフやイラストを作成するグラフィック・レイヤー，英数字や漢字JIS第一／第二水準をレイアウトするテキスト•レイヤーの3画面に大きく分けられ，

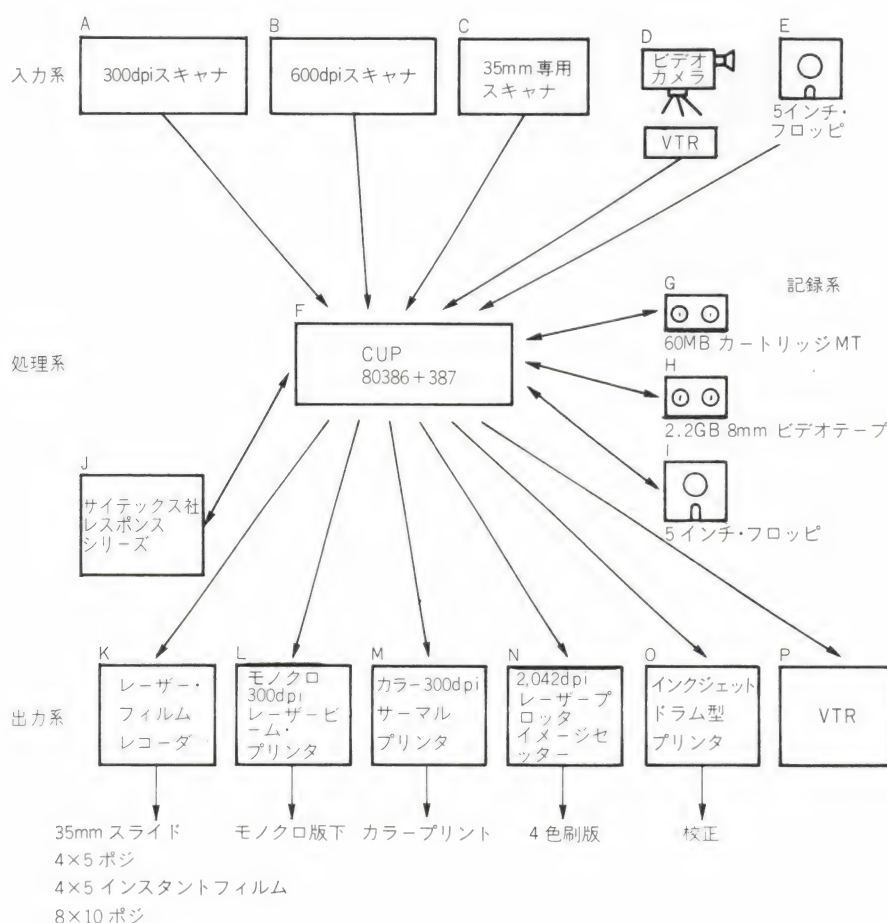



図2　イマプロ IPPS リタッチャーシステム図

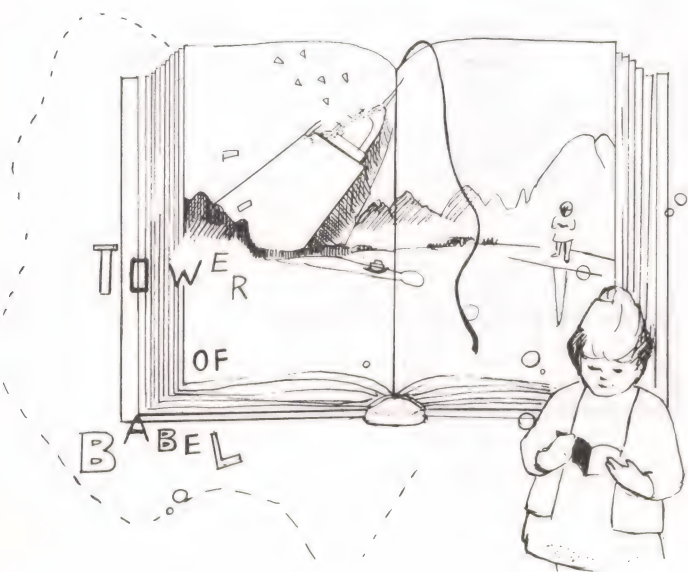
図3　合成する写真

図　4

画面どうしの重なりの順番や透過率はフレキシブルに変えられる。また，必要に応じてピクセル・レイヤーにすべて合成したり，素材を分けるように他のレイヤーにある情報（この情報は非常に小さいのでフロッピディスクに記録される）をフロッピディスクに記録して保存もできる。ピクセル・レイヤーの情報は約36Mバイトあるため，内蔵の60Mバイト・カートリッジMTか2.2Gバイトの8mmビデオ・スプーラーに記録される。この部分だけでも従来のトータルスキャナのMT（約150Mバイト）に比べてはるかに安価でスピードが速く，大容量である。また，クロック20MHzの32ビットCPU本体のメモリは18Mバイト，内蔵のハードディスクは300Mバイトと，パソコンでありながらパソコンのレベルをはるかに超えている。インタフェースもGP-IBをはじめSCSIなど今後の通信や大型システムとのリンクに必要なインタフェースは標準装備になっている。

出力系はレーザーにより鮮明忠実なフィルム出力のできるレコーダ，簡単な版下を作成できる300dpiレーザービーム・プリンタ，300dpiカラー・サーマルプリンタにはCMYKの別々の情報を送ることにより，カラーラフ，カラーカンプの出力を可能にし，イメージセッターといわれる2,032dpiのレーザー・プロッタからはそのまま印刷に使用できる4色版が出力される。フルカラー・インクジェット・プリンタからは最大60×60cmサイズのプルーフ出力が300dpiで可能で，VTRには768×512に間引きされた画像が録画可能である。また，大きな特徴として，従来のトータルスキャナであるサイテックス社のレスポンス・シリーズと双方向のハンドシェイクが可能など，どの点をとっても非常にユニークなシステムとなっている。

さて，システムの説明はこれくらいにしていよいよ作業に入ろう。まず，ビジュアルプランは「現代のバベルの塔」というテーマで作成する。合成する写真は3点，中国桂林の風景とク

◉図5〜7はCOLOR IMAGES参照。

ライスラービルと本を読む少女の3点である(**図3**)。そのビジュアルを切抜き合成し，レタッチするわけだが，それに加えてコンピュータ側で旧約聖書を表す輪郭のグラフィックと世界中の言葉が生まれたとされるさまざまな文字を書き加え，さらに特殊効果として崩壊するクライスラービルのレタッチ，川面に映るビルの影には少女が，少女の影にはビルがという不思議の表現，左下の少女の下半身とクライスラービルの屋上には砂目処理を加え，文字には透過率を別々に与えてビジュアル効果を上げることでラフを決定（**図4**），いよいよ入力に入る。

オリジナルはすべて4×5ポジなので，600 dpiスキャナを使用した。このシステムではスキャナ個体の色の読み癖をソフトウエアが自動補正し，オリジナルに最も近い状態で入出力できるようになっているが，カラーバランスを任意に狂わせることもできる。また，エッジ部分のシャープネスも5段階選ぶことができ，入力時にクライスラービルは最もシャープネスをきかせた。まず，ピクセル全体にベースカラーをペイントし，桂林の風景をトリミングしてレイアウトした後，クライスラービルを斜体にしてマスクを切り(入力時にマスクが切れる)合成(**図5**）する。次に少女にもマスクを切り，右上に小さく左下に大きく合成，各マスクの周囲を自然になるようにレタッチする。そして，川面に映るビルや少女をおのおのの素材から変形，透過合成し，少女の下半身とビルの屋上に砂目処理を行った。ビルは崩壊させなければならないので，屋上の一部を消し，鉄骨を書き加え，破片を適当にレイアウトし，シャドーをつけてそれらしく見せる。次にグラフィック・レイヤーに移り，聖書の輪郭や本の中開き部分，折り目などをイラスト的に描く(**図6**)。最後にテキスト・レイヤーに移り，各国の言葉をレイアウトし，それに透過率を与え，最後にタイトルの「Tower of Babel」をドロップシャドーで入力し，全体をピクセル・レイヤーに送ってビジュアルを完成，4×5でフィルム出力を行った（**図7**)。

この原稿執筆時に残念ながらレーザー・プロッタによるイメージセッターが日本に到着しておらず，4×5フィルム出力としたが，手元にはカナダより150線の出力サンプルが届いており，この原稿が発表されるころには，A3，175線4色刷版出力が日本でも可能になっているはずである。

これからのプリ・プレスシステム

当然，周辺機器のローコスト化と日本製のパソコンでのプリ・プレスは可能になろう。また，今後の大きな問題として漢字の書体の問題がある。現在，通産省が率先して書体開発を進めているが，書体メーカー大手の写研，モリサワ両社のもっと前向きで寛容な対応が望まれる。本当の意味でのページ・レイアウトはそれからであろう。また，1990年にスタートする予定のISDNという高速デジタル通信網は画像通信に大きく弾みをつけることが必至であり，それが可能になると他システムとの通信が簡単になり，日本国内外を問わず画像通信ネットワークが構築されるだろう。本当の意味で平成元年はパソコン・プリ・プレスシステムの幕開けの年となろう。

(79ページより続く。)

パーソナルコンピュータの限界と可能性を十分認識し，実際のデザインワークに対応できるシステムとして，当社の選択は正しかったと今思っている。

おわりに

CGデザイナーは2つに分かれている。

一方は，過去の技法をCGに当てはめようとして，エアブラシの機能やマスキングの能力の充実を要求している。

もう一方は，今までのデザイン技法からの決別を宣言し，絵筆をキーボードへと完全に移し，これまでにない技法を開拓している。しかし，それは過渡的様相であって，例えば車社会には馬の手綱さばきは必要ないように，過去の遺物を引きずった技法は確実に消えていくであろう。否，キーボードをたたくことさえ過去の技法となる日が来るかもしれない。一部ではその芽はすでに育っている。

これらは，決して本来の目的を逸脱したものではなく，より純粋にクリエーティブであるための一つの変化にすぎない。

過去において手先の器用さがデザイナーの条件の一つであったが，今は構造的に思考する能力がより重要になってきている。過去数年の間にめまぐるしく変わったCGが，今後の数年間でどのように変わるかは誰にも想像できない。ましてや，10年，20年先など全くわからない。あえて予想するならば，CGが特別の造形活動ではなくなり，日常的なものになっているだろう。現在，CGが芸術たり得るかということが話題になっているが，常にその時代の最高の科学が最高の芸術を生むことを歴史が教えている。もし，CGが芸術たり得ないとするならば，それは過去の規範においてであり，価値観の変化に追い付いていけない小心者の寝言にすぎない。

グラフィック・デザインにおけるCGは，それが何者かがはっきりした段階で相当高い位置を占めるだろう。現在の段階では，過去のデザイナーたちがおそるおそるふれている状況であり，いまだに“CGらしさ”などという，CGを何か特別な技法であるかのように思っているデザイナーが多い。

近い将来，CGになれ親しんだ新人類がデザイン界の中枢を占めるころには，例えば，カメラマンによる撮影であるとか，イラストレータによる描画のように日常的な技法の一つになっているだろう。またそのころには現在，写植機程度の価格であるパソコンCGシステムが，小規模のデザイン事務所でも買える程度まで価格が下がっているだろう。現在，どこのオフィスにもワードプロセッサがあるのと同じように，グラフィック・デザインの事務所にCGシステムが置かれる日は近い…

特集2 建築・土木のCADとCG

建築や土木の分野は基本的に単品注文生産の世界である。そこでは従来の経験則や模型を用いた実験データをもとに設計を行っていた。出来上がった建造物を用いて設計結果の検証をすることは不可能である。

そこで建築・土木の分野においては，特に大手のゼネコンや建築設計事務所を中心として設計業務のCAE/CAD化を強力に推し進めてきた。経験的に得られたデータや模型実験の実験データを利用し，従来より設計計画業務をより高度化するために積極的にコンピュータを応用してきた。そして最近のコンピュータ・ハード／ソフトの急激な高性能化と低価格化によって，大手のゼネコンや建築設計事務所でなくとも利用できるようになり，小さな建築設計事務所にも普及しつつある。

近年，情報化の波は建築・土木業界にも押し寄せてきており，小さな建築設計事務所や建築現場の作業所へも急速にOA化およびFA化が進みつつある。その中で建築CADシステムの利用も普及し始め，最近は作業所単位でパソコンCADなどを導入し，施工図を作成しているところも見られるようになった。

機能設計だけではなく建築の意匠設計の分野にもCGシステムが普及しつつあり，設計以前の受注入札の際のプレゼンテーションにCGが有効利用されてきている。特に最近は，大規模なコンペではCGでプレゼンテーションをしないと工事を受注することができないという声も聞こえ，より高度なCGアニメーションによるプレゼンテーションも行われるようになった。CGの産業応用という点では，現在最も成功している分野の一つであろう。

今回の特集で掲載した例だけが建築・土木の分野の事例ではないことはもちろんである。例えば，今後のリゾートブームなどの高まりなどによって，大規模な水上構造物の建築や環境景観をうまく生かした大規模な沿岸開発などの土木計画も多くなることが考えられる。また，従来例のない開発計画も立てられることだろう。ここでもまたCADやCGなどのシミュレーション技術がおおいに活躍するはずである。

PIXELでは，今後シリーズ企画として建築・土木のCADとCGを紹介していく予定である。産業規模としても大変大きなものがある同分野でのCADやCGの今後のあり方は，大きな期待がもてる。今後の動向を期待するとともに，読者の中で面白い取組みをしている方がおいでなら，ぜひ本誌編集部までご連絡いただきたい。

建築設計事務所とCAD

建築設計事務所におけるCAD利用は，次の3点に集約されつつあるという。まず，平面図から3次元DBを基本とする設計図書へ，という設計業務の比重移行が実施段階にあること。次に，CADソフト／ハードの進歩により標準DB化が進み，デザイン面の重要性が現在以上に増すこと。最後に，多様なCGによる建築表現が想定される。本稿では，建築設計事務所におけるCAD化の流れを解説するとともに今後のあり方を考察する。

照明計画とビジュアル・シミュレーション

従来より照明計画の照明環境の予測・評価は，照度分布図や輝度分布図，またプレゼンテーション用にはパース図を利用しているが，照明の専門家以外には必ずしも有効なプレゼンテーション手段ではなかったという。本稿では，照明シミュレーションを実用レベルで行えるCGシステムの構築を実現し，利用している東芝電材の設計事例をもとに照明分野へのCGの最新事例を紹介する。

厨房設計におけるCAD活用事例 ―レイアウト設計に対するデータベース応用の実際

最近のホテルや複合施設などの発達には目覚ましいものがあり，その主要施設である飲食施設はますます多機能化されている。ここでは，厨房施設の専門業者として15年間にわたりホテルや複合施設などの飲食施設の設計業務を行っているSMI社の厨房施設の機能設計におけるCADの活用事例をもとに，厨房機器における設計の最前線を紹介する。

シミュレーション・ツールとしてのゴルフ場造成システム

建築・土木の分野で古くから実績をもつ淺沼組のコンピュータ・シミュレーション技術の中で，最も新しい分野である土地造成設計支援システムとゴルフ場造成設計支援システムの中から，ゴルフ場造成設計支援システムを例にあげ，土木分野のシミュレーション技術の最新事例を紹介する。

大成建設におけるCAEの一例 ―自然力に対する構造物の応答性状をさぐる

新宿副都心都庁新庁舎などに代表される超々高層建造物では，地震などに対する耐震の問題だけでなく，ビルの風に対する耐風の問題が大きな課題である。本稿では，新宿センタービルなどに代表される高層建造物の施工に実績のある，同社の超々高層建造物の耐震・耐風シミュレーションへの最新の動向を解説する。

特集 2　建築・土木のCADとCG

建築設計事務所とCAD

建築設計事務所のCAD利用は，事務所規模や設計業務内容によって多様化しており，今後5年間でCADコンサルなどの新しい形態が生まれようとしている。

中山　信二*

◉図1は次ページ。

はじめに

建築設計事務所におけるCAD利用は，平成元年に入り新たな展開を迎えようとしている。それは，概括的に表現すると次の3点に集約されよう。1つは，平面図を中心とする設計図書から平面図・断面図・立面図を統合した3次元データベースを基本とする設計図書へというように，設計業務の比重の移動がいよいよ実施段階に入ることである。2点目は，CADソフト／ハードの進展と相まって標準ディテールのデータベース化が進む結果，設計業務におけるデザイン面での重要性が現在以上に高まり，業務形態も大きな変貌を遂げることである。最後に，CADの中で特に技術的な進歩が著しいコンピュータ・グラフィックス（CG）の分野において従来のリアリティの追求といった単純で明快な目的にそった建築表現ではなく，さまざまな価値観が交錯する現代情報社会にあって，人間の知性（インテリジェンス）と感性（センス）の欲求にこたえる建築思想や輻輳する建築条件に合致したきわめて多様なCGによる建築表現が想定されることである（場合によっては2次元の図面化が不可能となり，従来の範疇では模型によって行われた設計者から施工者へのコミュニケーションがCGによって行われるかたちが一般的になる可能性もある）。

はじめの3次元データベースに関しては一部に疑似的な3次元データ互換ソフトも出現しており，平面からパース，平面図の建具シンボルと建具表との連動処理といったソフトウエアも実用段階に入ったといえよう。また，標準ディテールに関しても営繕協会といった公的建築物に利用されているデータが今年中にはCADデータベース化（フロッピ化）され，発売されるといった動きもあり，一方ではCAD業務経験の長い設計事務所においてすでにCAD図面の標準化が推進されている。最後のCGによる建築表現については，ある意味でリアリティの追求（質感・陰影・光沢）といった段階から，より開かれたCGでなくては表現できない人間の知性（インテリジェンス）と感性（センス）の欲求にこたえる個性豊かな建築表現の追求という段階に移行したといっても過言ではない。

以上のようなCADにまつわる状況は，他の業種と比べて近代化が遅れている設計事務所のOA化という観点に立つと，実はOA機器の導入期から普及期という第2段階に入っただけにすぎない。設計事務所が生産性と創造性という相反する機能を統合した高度な情報発信基地へと変貌を遂げるのは，次の段階すなわちテクノストレスに象徴されるOA化に伴う弊害を克服する環境整備という第3段階，ひいては情報のデータベース化が完了した先の段階，つまり入力（インプット）された情報（データベース）を加工して価値ある情報として表現（アウトプット）する設計者（人材）があらゆる意味で尊重される第4段階まで到達しなければならない。したがって，そこにいたる道はかなり厳しいものがあるといえるが，今回は第3段階から第4段階にいたる設計事務所の近未来像はどのようなものとなるかを考察したいと思う。以下に一般的な設計事務所の業務形態や内容の過去5年間の推移を概観し，併せて今後5年間に予想される変化を先取りして記述してみたい。あくまでもCADと設計業務という観点に立っての小論なため論旨の不十分な点が多々あると思われるが，読者の御寛恕を乞う次第である。

設計事務所業務の過去・現在・未来（図1）

■過去（1984年以前）

- 一部のアトリエ事務所や合理化が進んでいる一部の大組織事務所を除き，設計業務の大部分は所内で処理されている。
- 参照設計資料は書籍・図面・文書が一般的であり，外部から

*なかやま　しんじ　㈱現代建築研究所　設計本部　〒160 東京都新宿区新宿2-8-8

過去(1984年以前)

現在(1989年前後)

未来(1994年以後)

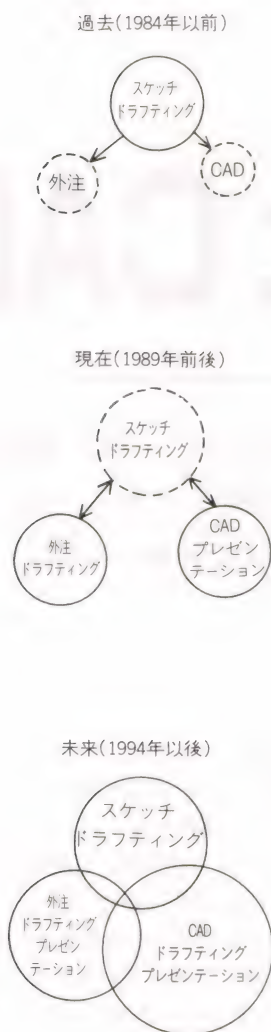



図1 設計事務所業務の過去・現在・未来

のノウハウは専門メーカーなどの一部に限定されていた。

- デザインのチェックやスタディは手書きパースと模型によって行われ，コンピュータの利用はごく一部に限られていた。
- CAD利用は大部分の事務所では日影・パースが一般的であり，設計業務に占めるウエイトは数%の域を脱していない。

■現在（1989年前後）

- 業務量の増大と建築物の複合化および小規模事務所の増加などの諸事情により，基本設計以外は外注で処理するケースが増加し，設計担当者が業務全体を掌握するのが困難な状況となっている。
- 欧米のような実施設計を担当するドラフトマン体制が未整備のまま外注（下請）事務所が増加したため，短期間での作成という観点からは小回りがきく半面，図面内容や精度に統一を欠き，設計主旨が徹底しないうらみが残る。
- 再開発にみられるように建築物の大規模化・複合化が進行する半面，人件費の高騰や人手不足により設計業務の合理化・効率化が急速に図られるため，設計の一部に創造性の欠如や質の低下を招くケースも目立つようになる。
- 本格的な情報化社会を迎え，OA機器の普及や設備条件の複合化が進んだいわゆる建築のインテリジェント化により図面表現の情報量が増大したため，改修工事を含む設計変更に多大なエネルギーが必要となってきている。
- 建築表現が社会の変動に伴いこれまでにない多様な展開をみせつつあり，従来の設計図(2次元図面)では表現しにくい建築が増加している（例えば，空気膜構造の集会施設など)。
- デザインチェックの一部にCADパースを利用する事務所が増加し，一部の設計コンペやプロポーザルにCAD図面を使うケースがみられるようになった。
- 企画から基本実施にいたるまでCADを利用する事務所はごく一部を除いて少なく，大部分の事務所はプレゼンテーションとしての利用が中心となっている。

■未来（1994年以後）

- 同一のデータベースによる企画から設計監理・メンテナンスにいたるまでの一括処理が一般的となる。特に，大規模高層建築や定期的な改修が予想される営業店舗などでの普及は急速に進行するだろう。
- 訓練されたドラフトマン・グループ(専門外注事務所)かCADチーム(オペレータやデザイナー，プランナーの別を問わず)が実施設計を担当し，従来の設計者の本格的な業務が企画・基本設計に集中する（監理業務ももちろん含まれるが，規模や施設内容によって複雑化する)。このような業務形態の変化に伴い設計事務所の組織も大幅な変更を強いられる。例えば，実施設計を外注するか社内で行うか，もしくは別会社をつくって委託するか，さらにはCAD業務を専門とするデザイン事務所との共同設計とするかなどについては設計事務所の規模・業務内容・所員のCADに対する習熟度などによって異なってくる。
- 標準ディテールのフロッピ化やCD（コンパクト・ディスク）化が普及し，その分だけ設計事務所は自社の標準設計図に安住することなく機能的でかつ個性あふれるディテールを競うことになる。
- 一部の事務所間でCADとデータ通信による共同設計などの利用が進み，設計のオリジナリティの問題をはらみつつ設計データの共有化が促進されよう。
- 建設業界のOA化が進み，従来型の設計業務では支障を来す状況が予想される（例えば，建築確認申請業務のOA化が平成4年に実施され,CAD図面だけでなくフロッピ形式の申請も可能となる)。
- 建築表現の3次元化が一部実現し，設計契約による建築主への設計図書の納品はフロッピやCDによって行われるようになるだろう。2次元の図面は施工や改修などの必要に応じ出力され，訂正箇所が入力され次第処分されることになる。
- デザインチェックは3次元CADによって行われ，単なる形態的・法規的チェックばかりではなく，照明・色彩・テクスチャ・視環境・空間密度・防災計画・景観計画・都市計画といったさまざまな設計条件をトータルに評価していくシステ

ムが開発されるだろう。

●参照設計資料はCADデータが一般的となり，相対的に職場環境が画一的となりやすく感性あふれる設計を生み出す環境作りがますます急務となろう。

● CAD利用はドラフティングも含めた総合的なものとなり，ある意味で図面訂正などの手間が全体の工程を左右することになり，企画・基本段階での設計者の価値判断が業務全体に与える影響は大きくなる。

CAD利用の過去・現在・未来(図2)

■過去（1984年以前）

●コンピュータ導入の事務所は少数（10％以下）であり，コンピュータ利用も汎用コンピュータを使った構造計算が主体で，CAD利用は日影・パース・収支計算にとどまる事務所が多かった。

● 16ビット・パーソナルコンピュータ（パソコン）が出始め，従来のホビー的要素の濃かった8ビット・パソコンから業務用コンピュータとして利用する事務所が出始める（ワードプロセッサ（ワープロ）とファクシミリが1982年ころから導入され始め，設計者，特にデザイナーのOA機器に親しむ下地を形成した功績は計り知れないものがある）。

■現在（1989年前後）

●過半数の事務所がコンピュータを導入（70％以上）し，特に30人以上の中堅事務所では導入済みのところが多いと思われる。利用形態は事務所の規模や業務内容によって多岐にわたりとても概括できるものではないが，大部分はデータベース志向の本来的なCAD図面のレベルではなく，試行錯誤的要素と合理化・先行投資的要素の強い清書用のドラフティング・マシンの域を脱しないケースが多いと思われる。

● CADが大規模プロジェクトのプレゼンテーション手段として定着したが，ハードウエア／ソフトウエアともコストが高く，利用は一部の組織事務所に限られている。

●パソコンCADがハードウエア／ソフトウエアの性能アップと低廉化により急速に普及しており，一部の事務所では基本から実施にいたるまですべてCAD化しているところも出始めた。

● CAD化を推進している事務所の平均的な姿は，基本・実施設計の中のCAD化が比較的容易な平面や立面といった一般図に限られ，その他は従来の手書き図面を併用するかたちが多い。提出期限などの制約がある場合は軽微なCAD図面の訂正を手書きで行うことも多く，ひいては図面の食い違いや表現上の曖昧さが付きまとい，CAD化のメリットも少ないといえよう。

●一部にCADのネットワーク利用やビデオ化など，CAD化のメリットを生かした利用形態も出始めている。

● 32ビット・パソコンが出始めて最大のネックであったパソコンCADの処理速度が大幅に改善されるばかりでなく，分散型端末のためホスト・コンピュータによるCAD処理で起きる能力ダウン現象からも解放され，効率の良いCAD業務が可能となってきた。それに伴いCAD端末の台数も複数に1台から1人1台の体制に移行していくだろう。

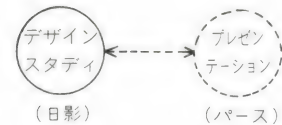
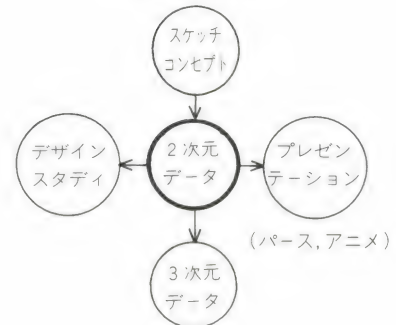
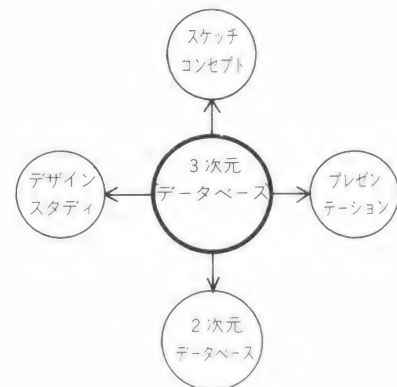



図2 CAD利用の過去・現在・未来

●手書きと比較して小回りがきかない象徴的存在であったプロッタなどの出力機器(密度の濃い図面の場合，打出しに1時間近くもかかっている）が大幅に改善され，特に価格が数十万クラスのレーザー・プリンタによる出力が可能となって機動力が高まってきた（詳細図の打出しでも数分以内で可能となり，所内でのチェックや基本設計段階では十分活用できる）。

■未来（1994年以後）

●大部分の事務所（90％以上）がコンピュータ導入を完了している。手書き併用CADが一般的であるが，一部の事務所では

	CPUの能力	外部ディスクの容量	入力装置	出力装置	図面作成時間	ケーブル伝送速度	図面1枚伝送速度	他のシステムとの互換性	グラフィック機能	コスト・パフォーマンス
過去 約5年前 （1984年頃）	8ビット 16ビット	1MB 10MB フロッピディスク 磁気ディスク	キーボード	プロッタ	3～5日	0.1Mbs 同軸 ケーブル	2～10 時間	ほとんどなし	640ドット 8～16色	1 一定の図面を作成する労力とCADシステム価格との比率で5年前を1と仮定
現在 （1989年前後）	16ビット	20MB 40MB 磁気ディスク	マウス キーボード	プロッタ ハードコピー 写真	1～2日	1～10 Mbs 一部光 ケーブル	0.2～1 時間	AutoCAD を中心に 一部データ 互換可能	1,120 ドット 196色	10～100 単一図面の作成・出力速度が向上
未来 約5年後 （1994年頃）	16ビット 32ビット	200MB 600MB 光ディスク （CD化）	イメージ ボード スキャナ マウス キーボード	静電 プロッタ ビデオ CRT 各種コピー	数時間 1日	10～ 100Mbs 光 ケーブル	0.02 ～0.1 時間 （1分 ～数分）	大部分の ソフト間 でデータ 互換可能	1,120～ 約4000 ドット 1,670万色	100～1,000 データベース利用により，複数図面の作成・出力速度が向上
備　考		図面1枚 約0.05～2MB			平面図 簡易外観 パース	ネットワーク利用	インターフェースソフトの処理時間を除く	2次元間だけでなく3次元データとの互換も可能		前提条件 システム価格の10年間のコストを一定とする 処理性能／システム価格 処理性能には処理速度の他にネットワーク化による効率アップやデータ精度を考慮

図3　建築CADの過去・現在・未来（パソコンCADのハードとソフトを中心に）

3次元データベースによる100％CADが実現しているだろう。

- 住民説明やデザイン・コンセプトの意志決定の有力手段としてCGが活用され（ビデオ・パースなど），CG業務を専業とする設計事務所（？）も確立される（現在でも関西空港などの例にみられるように，一部にCGが活用されるケースがなくはないが，設計業務の契約事項に含まれたり，設計料としてCADの予算が組まれるケースはきわめて少ない）。
- 標準ディテール・積算データ・雑誌発表作品などのフロッピ化・CD化が普及し，建築行政上でも建設省の積算データや確認申請データのCAD化が実施され，住宅産業を中心として図面のCAD化が急速に広まるだろう。
- デザイン発想のツールとしてのCAD利用の普及がアトリエ事務所を中心として促進される。
- CADのデータベース的利用が一層促進される。利用形態はさまざまであるが，想定されるものとしては次のものがあげられよう。小規模事務所間，建築・設備・構造・インテリア事務所間，事務所―メーカー間，事務所―建築主間，事務所―現場間，事務所―協力事務所間など。（図3）

設計事務所におけるCAD利用の概要

前述したように設計業務とCAD化は互いに影響し合いながら多様な展開を示すと思われるが，ここで今一歩想像力を駆使して現在から5年後にいたるまでの建築設計事務所のCADを中心とした将来像を描いてみたい。

まず，多様で複雑極まる設計事務所の組織形態を類型化するとすれば，下記のものが想定されよう。比較しやすくするために次の5項目について分類してみた。

①CRT端末1台当たりの設計スタッフ数
②CAD入力の担当部署（所内入力／外注入力）
③CADの利用目的（デザイン志向／効率志向）
④CAD入力担当の職種（デザイナー／オペレータ）
⑤勤務体制（レギュラー／シフト／人材派遣）

- タイプ1：組織事務所型
  - 大型CPUとパソコンを併用
  - ①については急速に減少する（部によっては1人1台）
  - ②～⑤については併用が多い
  - タイプ2～5の性格も組織の規模・業務内容・体制などにより併せ持つ
- タイプ2：アトリエ事務所型
  - パソコン主体
  - 近い将来1人1台体制となる
  - 基本設計（デザイン）についてデザイナーが所内で入力し，シミュレーションなどにも利用し，実施は外注を併

	事務所所員数	端末1台当たりの所員数	CAD入力担当部署	CADの主な利用目的	CAD入力担当の職種	勤務体制	CADシステムの概要
組織事務所型	100人前後 ～数百人	数人～10人前後 近い将来CAD部門で1人体制	所内にCAD部門を設け入力 効率の悪い入力は外注	設計監理・コンサル業務の全体にかかわる総合的なCAD利用を図る	デザイナーとオペレータ 所内研修体制の整備が前提	レギュラー 人材派遣（オペレータ）はシフト制	汎用からミニコン／パソコンを業務内容に応じ使い分ける
アトリエ事務所型	10人前後～ 100人前後 中小規模事務所	2人～10人 近い将来1人1台体制となる	プレゼンテーションとシミュレーションは所内実施などは一部外注	デザイン中心 手書き・模型などCAD以外のメディアを併用	デザイナー	レギュラー他	パソコン主体 CG能力の高いシステムを付加
ドラフティング事務所型(プロダクション型)	10人前後～ 数十人 中小規模事務所	1人以下 効率を高めるため，1人で数台の端末を処理する	100％所内入力 協力事務所との共同設計もある	効率的なドラフティングを優先	オペレータ	シフト制が多い	パソコン主体 効率の良いEWSも併用
OA推進事務所型	数十人～ 100人前後 中規模事務所	1人～2人	プレゼから基本実施を所内入力 業務量に応じ一部外注	デザインと効率的ドラフトのバランス重視	デザイナーとオペレータ 業務量が偏るため調整が必要	レギュラーとシフト制の併用	パソコンとミニコンCADシステム併用
外部依存型	数人～数十人 小規模事務所	事務所に外部通信用の端末数台程度	100％外注 利用可能なデータは所内訂正	デザイン優先 効率的なドラフティングを外注	外注事務所	小規模なためイレギュラー OA機器を活用した外部通信	データ通信に必要なパソコン程度 CADシステムの共同利用を図る
CADコンサル型	数人～数十人 小規模事務所	1人以下 所内に外部貸出用端末も常備	CADの研修・指導が中心 契約内容により共同設計もある	デザインとドラフトを含めたシステム的発想による利用	システムエンジニア的立場	イレギュラー	パソコンから高度CG能力をもつシステムまで導入 一部研修・貸出用

図4 建築設計事務所のタイポロジー（CAD利用の観点からの分類）

用するかたちが多くなる

- タイプ3：ドラフト事務所型
  - CADによる効率の良い図面作成が主業務
  - 外部からの委託によるかたちが多い
  - 多数のオペレータによる入力が中心で1人1～2台体制
  - シフト制や人材派遣を活用
- タイプ4：OA推進型
  - CAD以外の所内業務もOA化を推進し，少数精鋭型の事務所を志向
  - 近い将来1人1台体制となる
  - 所内入力を基本とするが，場合により外注やデータ通信による共同設計も志向する
- タイプ5：外注依存型
  - CAD化・OA化を必要最小限に抑え，従来型の設計業務形態の長所を温存していこうとする。したがって基本的にCAD業務は外注に依存する。場合によっては事務所どうしのCADシステムの共同利用を図る
- タイプ6：CADコンサル型
  - 設計業務は直接行わず，CADによる設計を実施する場合のノウハウを提供する一種のコンサルタントサービス会社。今後出現が予想される （図4）

■タイプ1：組織事務所の事例（内容は多少抽象化してある）

1) A設計（内容は建築CADに限定，他の事務所も同様）
   - 全国的なブランチをもつ大規模事務所
   - 東京と大阪のCAD業務のデータ互換を図り，プレゼンテーションだけでなく一部の実施設計にもCADを利用している

〈東京事務所〉

- ミニコンCADシステム（GDS）端末23台
- プレゼンテーション中心（パース，アニメ）であるが，最近はドラフティングにも力を入れ始めた
- 設計情報室（4名）はソフトウエア開発・研修が主体で，オペレートは設計スタッフが行う
- 設計室他にパソコン，ワープロ約200台（AutoCAD約20台含む）以上保有するが，各セクションごとの個別利用にとどまる
- 新入所員全員にCAD研修を実施しているが，1989年から研修時間を20時間から100時間に延長して行う予定

〈大阪事務所〉

- ミニコンCADシステム（GDS）端末7台

図5　バスの事務所内部

現代建築研究所設計室

- パソコン（AutoCAD）端末58台
- パソコンCADによる実施設計も一部実施
- 入力は人材派遣の女性オペレータによる
- GDSとのデータ互換可能

2)　B建築事務所（東京）
- 数百名の設計スタッフをかかえる大規模事務所
- ミニコンCADシステム（GDS）端末6台
- プレゼンテーションとドラフティング
- 所内にパソコン，ワープロ約40台（CAD対象外）
- 設計スタッフ（約120人）の30％がCAD研修完了
- オペレートは設計スタッフが行う
- ドラフティング用基本データライブラリ作成

3)　C設計事務所（東京）
- 数百名の設計スタッフをかかえる大規模事務所
- ミニコンCADシステム（GDS）端末13台
- YHP 9100端末4台（設備CAD用）
- システム開発とオペレートの一部を担当する別会社（所員11名）を設置
- オペレートは専任オペレータと設計スタッフがペアで行う
- 新入社員のCAD研修は80時間
- 所内にパソコン，ワープロ約40台（CAD対象外，ラップトップは除く）
- 手書き図面の外注比率が高いため比較的ドラフティング分野が遅れている。

■タイプ2：アトリエ事務所型の事例

1)　Dアトリエ
- 所員数約30名で前衛的なデザインで著名
- パソコン（DRA-CAD）端末15台
- プレゼンテーションとドラフティングをケースバイケースで使い分ける
- オペレーションは基本的に設計スタッフ（デザイナー）が行うが一部外注化している
- CAD研修は特に行っていないが，最近自発的にCAD入力する所員が増加

2)　E設計事務所
- 所員約50名で公共施設のデザインが多い
- パソコン（DRA-CAD）端末6台
- プレゼンテーションとドラフティング
- 建築を構成する部品などのデータベース化を図りデザインに反映させる
- オペレーションは設計スタッフが行うが一部外注化を図っている

■タイプ3：ドラフティング事務所型

1)　バス（**図5**）
- 所員約15名でプレゼンテーションからドラフティングまで一貫したCAD化を実施している画期的な事務所
- DRA-CAD端末約30台（一部貸出用）
- 端末1人に約2台（入出力業務の効率化）導入
- 現在はデータ入力代行業務が多いが，将来はデータ作成

図6　CADセンターのアニメーション作成システム(視点・軌跡などのデータを入力すれば，端末の方(テクトロニクスT4237)で画像処理を行う。)

図7　手書き用の製図板(A1との併用であるが近い将来,出力機器を増設し，製図板はテンポラリーもしくはスケッチ専用に用いることになるだろう。)

図8　現代建築研究所CAD室(ミニコンとパソコンを併用し，データの互換性は確保している。簡単な休憩もここで行うため,外部の景色が眺められる位置を選んである。1989年には増設して端末数が合わせて10台となる。)

◉図9〜18はCOLORIMAGES参照。

からCADによる設計を含む総合的な中規模事務所を目指している。したがって，従来のドラフティングという言葉から連想する実務一本槍ではなく，より広くとらえたプロダクションという概念に近い存在となろう。場合によりCADコンサル的な業務も担当している。

- オペレーションは設計スタッフ（1人）とオペレータ（2人）がチームを組んでシフト制の勤務体制をとる
- 国内数箇所や海外の協力事務所との間でデータ通信を利用した共同設計を実施している

2）CADセンター（**図6**）

- ARCヤマギワから1987年に独立したCAD設計組織
- ミニコンCADシステム（GDS）端末12台
- CADデータの作成代行からCAD設計にいたる広範囲な業務担当（大規模プロジェクト中心）スタッフ数約15人
- アニメーション作成の外注も担当

■タイプ4：OA推進型

●現代建築研究所（**図7〜18**）

- 所員数55名。昨年創立40周年を迎えた典型的な実務型中堅事務所であったが，10年前からコンピュータを導入して設計業務のOA化を模索し，昨年全所的なOA化に踏み切った。
- ミニコンCADシステム（GDS）端末3台
- パソコンのネットワーク化を図りパソコンCAD（Mr. Jelly），ワープロ，在席管理，表計算他の多機能端末を約40台設置
- GDSとパソコンCAD間でデータ互換

図19は114ページ参照。

- プレゼンテーションとドラフティングの併用
- オペレートはGDSをオペレータが，パソコンCADをオペレータと設計スタッフが担当
- 現在所員2人に1台体制から1.5人に1台体制に移行中。近い将来1人に1台設置予定
- 昨年は基本･プレゼの比率が高かったが，1989年度は実施の比率を高めていく予定(88年度は全体の約30％程度)。

■タイプ5：外注依存型

●アークス建築研究所（**図19**）

- 所員約4〜5名の小規模事務所
- 所内ではCADの入出力は行わない
- モデムを使用して外注事務所（CAD担当）とデータをやり取りして図面のチェックを行う（DRA-CAD）
- 複数の小規模事務所間でのCADシステム共同利用を模索中

（114ページへ続く。）

特集 2　建築・土木のCADとCG

照明計画とビジュアル・シミュレーション

当社の照明計画支援CGシステムとその利用事例を紹介し，物理的な裏付けに基づいた照明環境の可視化にCGを利用する効果と問題点，課題について述べる。

菊地　壮一*
鹿倉　智明**
植田　慶幸***

従来の照明計画とCG

■従来の照明計画

照明計画においては，従来から照明器具の光学設計や照明設計に計算機が利用され，照明環境の予測・評価には照度（illuminance）分布図（**図1**）や輝度（luminance）分布図などが，プレゼンテーションにはスケッチ画（パース）が利用されている。これらの手段は建築設備設計者といった専門家に対してはそれなりに有効であったが，専門外の人や施主，ユーザーに対しては必ずしも有効なプレゼンテーション手段ではなく，よりわかりやすいプレゼンテーション，視覚化されたプレゼンテーションが求められていた。

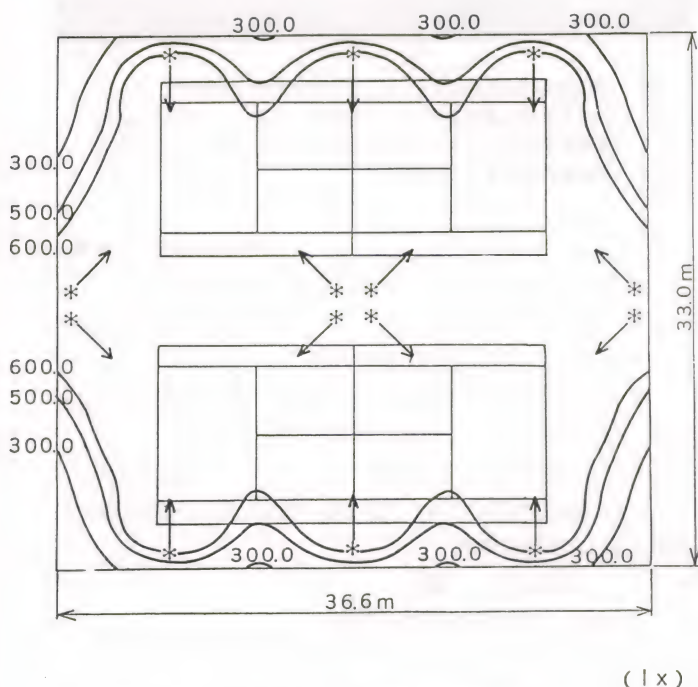



図1　テニスコート面の照度分布図例

■従来のCG

さまざまな分野においてシミュレーション結果の可視化の必要性からコンピュータ・グラフィックス（CG）が注目され，盛んに利用されてきているが，ソフトウエア，ハードウエアの発展に支えられて，その期待された効果を発揮しつつある。

しかし，一般的なCGシステムは疑似的な光源にある強さを与えてレンダリングするものであり，われわれの求める照明シミュレーションが行えるものではなかった。また，研究レベルでは照明された空間の色や明るさの分布をCGで可視化することが試みられてきているが，照明環境を物理的な裏付けに基づいた精度レベルと描画レベルを備えたわれわれが実用レベルで使用できるCGシステムは全くなかった。

こうした状況の中で，われわれはインテグラの協力を得て，照明シミュレーションを実用レベルで行えるCGシステムの構築を実現し，利用を始めているので，システムの特徴と利用目的や事例などを紹介する。

照明計画支援CGシステム

■照明におけるビジュアル・シミュレーションの目的

CGによる照明シミュレーションに対する要求のレベルについては，色や輝度分布がどのくらい忠実に表されるべきか，また形状入力作業を含めた画像作成時間やコストはどうあるべきかなど，利用の目的に応じていろいろな要求水準があるように思う。われわれは，色，照度分布，輝度分布などの情報をできるだけ物理的に忠実に求め，これらの情報を相対的な静止画像としてCRT上に表示することを目的としている。

そこで，予測しようとする物体の輝度分布に影響する要因と

*きくち　そういち　**しかくら　ともあき　　東芝ライテック㈱　研究所　〒237　神奈川県横須賀市船越町1-201-1
***うえだ　よしゆき　　同　セールスエンジニアリングセンター　〒108　東京都港区三田1-4-28

して次のものを取り上げた。

(1) 光源(照明器具)

● 取付位置，エーミング角
● 配光特性
● 分割数(点光源としてみなせる程度に光源を細分割する)
● 光源色(3刺激値R，G，Bに比例した数値)

(2) 視点の位置と方向

● 面素の位置と方向
● 面素の色(R，G，Bで指定)
● 面素の反射特性

光源データについては，これまでに使用，蓄積してきたデータを利用できるようにパラメータなどを共通のものとした。特に，照明器具の性能を表す配光特性データについては，当社で扱っている照明器具の中から1,300機種を現在データベースとして用意してあるが，随時追加していくことができる。

エーミング角は照明器具の照射方向であり，従来からわれわれが照明設計で扱っているものと同様の定義が可能である。

光源色は，蛍光ランプ，メタルハライドランプ，高圧ナトリウムランプ，低圧ナトリウムランプ，水銀ランプなど，使用するランプによる照射面の見え方の違いが表せるように定義できる。

■システムの利用目的

照明計画を広く支援することを目的としている。具体的には次のような目的を考えている。

(1) 光環境の忠実なシミュレーション

建築設計，建築設備設計，インテリア・デザイン，電気設備設計などにかかわる人々が，照明効果を計画の段階で視覚的に確かめることができる。特に照明空間を構成する材料の質や色，光源の種類(光色)，器具の配光特性の差異による照明環境の違いが視覚的に確かめられる。また，これは重要なことだが，室内など閉じられた空間の相互反射光をシミュレーションすることができるのも，本システムの特徴の一つである。この他，われわれは本CGシステムを照明設計のみならず，照明器具の設計においても照明器具の設計の意図，効果の確認のために使用している。

(2) 景観とインテリアのシミュレーション

画像合成，カラー・ペイント機能などにより，照明器具と周囲の環境との意匠上の調和を検討できる。

(3) プレゼンテーション

上記で作成した画像により，現実感に富んだビジュアル・プレゼンテーションが可能である。

■システムの構成

ハードウエアの構成を**図2**に示す。本システムは16ビット・パーソナルコンピュータを入出力I/Oコントローラに，32ビット・アクセラレータを照明計算と画像生成に使用する専用機である。

ソフトウエアの構成は，①照明シミュレーション・プログラム，②対話式3次元形状入力プログラム，③2次元カラー画像処理プログラム，などからなっている。

活用事例

当社では本システムを，技術営業支援を行うセールスエンジニアリングセンター（略称，SEC）で利用しているが，専用機であることを生かして全国展開を行っている。現在，東京・本社と大阪・西日本支社にフルシステムを設置し，通常の照明物件に活用しているが，他の地域においてもシステムの導入を随

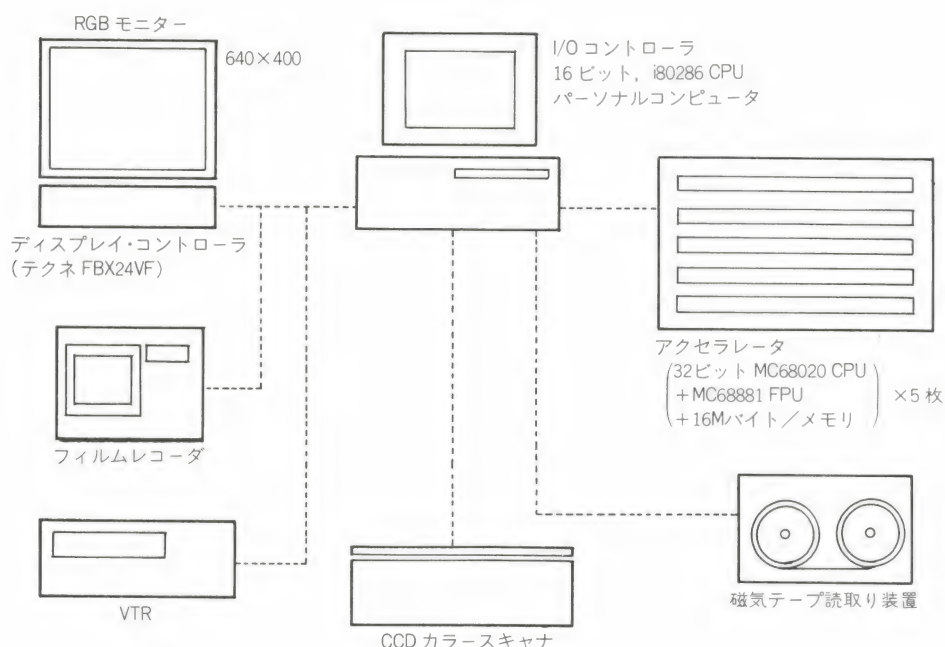



図2 照明度計画支援CGシステム構成図

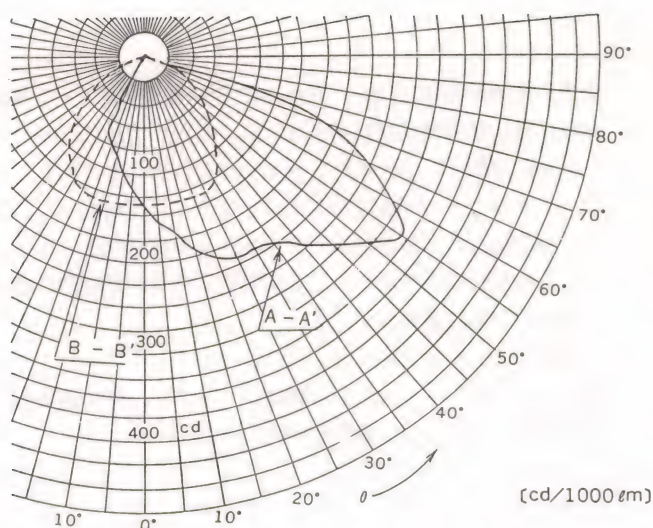



図5　テニスコート専用照明器具配光曲線図（タイプ1）

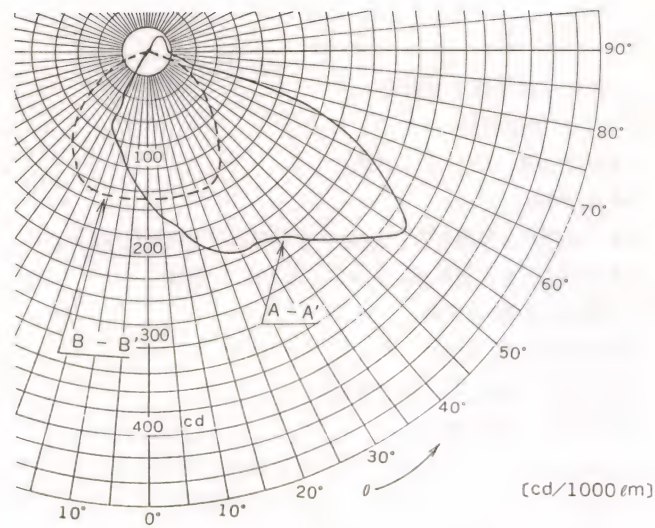



図6　テニスコート専用照明器具配光曲線図（タイプ2）

時行っていく予定である。特に従来の方法ではプレゼンテーションできない照明物件や，特殊な照明効果が要求される場合の照明効果の確認に使用している。

以下，CG を照明シミュレーションに適用した事例をいくつか紹介する。

■テニスコート照明

本照明シミュレーションは，当社の SEC が実際に某テニスコート照明物件の提案に使用した事例であるが，照明器具の照明効果を説明するために作成したものである。

1kW メタルハライドランプを使用した2種類のテニスコート用照明器具24台をそれぞれテニスコート3面に高さ6 m で水平に取り付け，プレーヤーがコート上から見た場合のシミュレーション例（**図3，4**）である。使用した照明器具の配光特性を**図5，6**に示す。

このシミュレーションでは照明器具上方に光が照射されるか否かの違いによって，どの程度ロビングボールの見え方に違いが生じるかをシミュレーションしている。照明器具上方への光がない場合（図3）には，照明器具よりも上方へ上がったボールには直射光が当たらず，プレーヤーからボールが見えなくなってしまう。これを上方への光がある照明器具を使用した場合（図4）には，上方に上がったボールにも光が当たり，ボールが確認できる。

従来の水平面照度，鉛直面照度といった断片的な評価データでは，コート面の照度（明るさ）が十分かどうか，あるいは均一に照明されているかについての情報を得るにとどまっていたが，CG を用いて照明シミュレーションを行うことにより，この例のように照明空間における照明器具の特性による照明状態の差異，見え方の差異を容易に視覚的に確認できる。しかも，照明の目的，使用条件などさまざまな条件による照明状態をテニスコート面，ボール表面などあらゆる被照面について同時に評価でき，従来の方法だけではなかなか把握できなかったスポーツ照明空間全体における光の方向性や，数値データでは判断しきれないスポーツ照明空間の質といったものの評価が，比較的容易にわかりやすいかたちで行える可能性があることがわかる。

■街路照明

ここでは，街路照明環境の研究のために作成した照明シミュレーション例（**図7〜9**）について述べる。

街路照明では路面を明るく均一に照らすだけでなく，防犯上の理由から歩行者の姿，表情が見てとれる明るさが必要になってくる。これらの目的を満たすためには照明器具自体の性能の他に，器具の取付け高さが重要となる。そこで，これらのファクタがどれだけ照明環境に影響するかを確認するために，照明器具の配光特性と取付け高さを変えた場合の街路照明空間の変化の様子をシミュレーションしている。

図7の照明環境は，照明器具の下方を重点的に照明する器具を高さ 5m，取付け間隔 25 m で設置した場合である。路面に明暗のむらが生じ，照明器具の真下に立った人はシルエットとなってしまうなど，非常に好ましくない照明環境となっている。

これに対して，図8では路面を均一に照らすだけでなく人を明るく照らす照明器具を用い，取付け高さも 7m と高くしたことによって照明環境が改善され，歩行者にとって良好で防犯上好ましい照明環境が得られることがわかる。

また，道路部分の照明に高圧ナトリウムランプを，他の部分ではメタルハライドランプを用いた場合の照明環境の変化を図9に示している。この例のようにランプの種類，光色を取り扱うことは，照明シミュレーションでは重要であり，不可欠である。

■室内照明

床据置き型間接照明器具（250 W メタルハライドランプ）を4灯使用した場合（**図10，11**）の室内の照明状態を示している。

このシミュレーションでは照明器具からの直射光は天井と壁にだけ入射し，床面には全く入射していない。従来の CG では，疑似光源を設定しないかぎり床は暗くなるはずである。ここでは鏡面，拡散反射光を扱った例として，天井面から床面への光の入射回数を最高2回となるように反射回数を設定している。つまり，反射回数を3回としている。正確な床面の照度値を得るためにはまだまだ追跡回数が不十分であるが，おおよその室内の照明状態を把握するには十分であることがわかる。もちろ

◉ 図3,4,7〜11はCOLOR IMAGES参照。

ん,計算時間などコストに制限がなければ,さらに反射回数を増やして精度を上げることが可能である。

従来の照明計算では室内など計算対象となる空間における障害物が考慮されていなかったため，最近のオフィスのようにパーティションを多用した空間において施工後の状態を事前に評価,検討することはほとんどできなかった。CGを利用することにより，この事例のように照明器具などからの光がパーティションによって遮られて机上面が暗くなる様子や,微妙な影の状態など，従来にないさまざまな評価要素を含めた総合的な検討が行えることを示している。

図11は図10で示した照明環境を照度の帯で示した例であるが，物理的な裏付けに基づいて計算しているので，シミュレーション画像に対応した照射面の法線照度を表示することができる。

このように，本システムにおいては相互反射を扱えることから，あらゆる照明方式に対してその照明効果と照明環境の違いを検討することができる。例えば，図10は間接照明方式による場合のシミュレーションであるが，天井埋込み器具によって照明された空間をシミュレーションすることももちろん可能である。

これらの例のような相互反射光を取り扱った室内の照明シミュレーションを実用レベルで可能としたシステムを使用している例は，国内の照明メーカーでは初めてであり，海外においてもほとんどみられない。

今後の課題と問題点

■データ入力の問題

われわれは本CGシステムの位置付けを，照明計画を支援するためのものとしており，建築データなどの3次元形状データの入力そのものを本来の目的とは考えていない。もちろん，本CGシステムにおいても必要最小限の3次元データ入力を行うことはできるが，照明シミュレーションを行う場合，形状入力作業が実際の作業工程中の6〜8割を占めてしまうのも事実である。

建設会社，建築設計事務所などにおいて，CADによって意匠検討などが行われ，形状データがすでに構築されていれば，これらのCADデータを必要とするデータに変換して使用することで目的とする照明シミュレーションについての作業に集中することができる。

われわれは，現在，エイ・アール・シー・ヤマギワのCAD「GDS」のデータの読込みを実現させているが，この他の建築CADとのインタフェースについても随時拡充していきたいと考えている。具体的には「INTERGRAPH」「ICAD」などのデータ変換プログラムの開発を行う予定である。

■計算時間の問題

シミュレーション結果としてカラー・モニター上に表示する情報は輝度（明るさ）の分布であるが，この輝度は入射してくる光の方向と強さ，物体の反射特性，視点の位置の相互関係によって定まる値である。特に視点の位置にかかわる部分が大きいため，視点の位置が移動するたびに可視面の輝度計算が必要になってくる。このことからもわかるように，照明シミュレーションに必要な計算は膨大な量に及び，室内の相互反射までを取り扱う場合にはさらに膨大な計算が必要となる。これをリアルタイムに行うことは困難であり，現在アニメーション化は行っていない。この問題に対しては，ハードウエアの高速化と低価格化を待たなければならない。

■表示輝度の問題

実際の空間における輝度を考えると，通常の事務室内においては蛍光ランプの輝度はおよそ1.0×10^4(cd/m^2)であり，影の部分などは数cd/m^{2}である。屋外などではこれ以上の輝度差がある。

これをそのままカラー・モニター上に表示することは現在ハードウエア的に不可能なので，照明シミュレーション結果としての輝度の分布状態をモニター上の輝度に変換する際には注意を要する。われわれは，実際の照明空間において知覚される輝度（brightness）分布がカラー・モニター上で知覚される輝度の分布に相対的に等しくなるよう輝度の比を対応させることを試みており，単に計算輝度を256階調に直線補間しているわけではない。したがって，従来のCG画像にない現実感が得られていると思う。

おわりに

われわれが照明計画を支援する目的で開発しているCGシステムの概要と，照明シミュレーションへの適用例を述べた。

システムは，従来の照度分布図や輝度分布図をもとにした照明計画を補い，より質の高い照明環境を設計する道具として有効なだけでなく，大多数の人に対して照明の効果を直感的にわかりやすく説明するための手段としても有効であると感じている。そのためにも，輝度分布や色彩をできるかぎり忠実に再現できるシステムを構築するべく，今後とも努力していきたいと考えている。

最後に，CGシステムの構築にあたり協力いただいた関係各位に，この場を借りてお礼申し上げます。

参考文献

1) P. Jansen,I.Lewin : "The Effect of Room Obstructions on the Calculation of Inter-reflected Components", 1982 IES Annual Conference
2) 西田,中前：「影を考慮した面光源による照度の計算とその表示法」，照明学会誌，68-2, pp.9-10, 昭和59年
3) 中前,西田：「コンピュータ・グラフィックスによる光のシミュレーション」, PIXEL, No. 51，pp.154-158, 1986.12
4) J.W. Gregory, M.R.Francis : "A New Technique for Computer Simulation of Illuminated Spaces", 1987 IES Annual Conference
5) D.L.Dilaura, D.P.Igoe, P.G.Samaras, A.M.Smith : "Verifying The Applicability of Computer Generated Pictures to Lighting Design", 1987 IES Annual Conference
6) D. D.Miller, P.Y.Y.Ngai : "On the Realistic Computer Graphic Display of Luminous Environment", 1987 IES Annual Conference
7) H.Kajiyama, S.Kodaira : "An Illuminance Analysis in Partitioned Spaces using Monte Carlo Method", 1988 IES Annual Conference
8) 菊地，金子，高橋：「照明計画支援コンピュータグラフィックスシステム」. 照明学会誌, 72-11, pp.14-18 昭和63年

厨房設計におけるCAD活用事例
――レイアウト設計に対するデータベース応用の実際

「図面を書くだけのCADならいらない」から開発構想を始め，RDBとのリンクを果たした業界初の厨房設計用CADシステムについて，その3年間の活用事例を紹介する。

金子　孝一*
寄口　文雄**
石松　昇***

はじめに

近年のホテルや複合施設などの開発には目覚ましいものがあり，主要施設としての飲食施設に求められる要望はマルチファンクション化してきている。ホテル，レストランなどの業務用飲食施設の機能設計（多くの場合，厨房設計とよばれる）も，オーナーからみたキャッシュ・フロー効率化の要として重要度が増してきてはいるが，こと日本ではこの厨房設計の重要度の認知が遅れたまま今日まできた感がある。いくら立派な建物ができても，飲食施設という「食を通じてのサービスが伴う空間と設備」は，欧米に比べ往々にして貧弱な場合が多い。

当社は15年間，この飲食施設の機能設計業務を国内・海外と受諾し，オーナー・オペレータの要望を飲食施設のコンセプト設定から実施施工監理までのコンサル業務を通じ，かつゼネコン，建築設計事務所，関連工事会社との打合せを重ねて最適な施設機能を提案し続けてきた。

この設計分野では，1982年ころアメリカでデータベースを取り入れて本格的なCAD稼働を成功させているが，当社は1984年より標準化を中心とした準備を開始し，独自の開発コンセプトを打ち立て，1985年には日本ユニシスの力を借りてソフトウエア開発に着手し，1986年3月には厨房機器データベース化の完了とともに業界初の厨房設計用CADシステムの稼働を成功させた。なお，多くのユーザーの要望にこたえて1988年からソフトウエアを，1989年から厨房機器データベースをそれぞれ外販開始した。

厨房設計CAD化への背景とコンセプト

飲食施設の機能は，建物全体のコンセプトおよび設備の全体計画の関連より議論され，あるべき姿が決定されるが，通常，厨房の機能設計業務は**図1**に示すように，このコンセプト設定から基本設計，詳細設計そして入札仕様書作成へと多くの打合せを重ねながら行われる。その際，必ず必要とされるのが図面や設計資料であり，打合せのたびに修正・訂正が伴う。その作業量は本来必要な思考(Think)の時間を奪い取るほどに多量であり，作業の質は設計意思の伝達が難しいほど入り組んでいる。

それは，厨房という機能そのものが複合機能（クッキング，プレパレーション，ウォッシング，サービングなど）であり，選定される機器も数百種，機種にいたっては数千種（同じ機器でもメーカー，仕様，価格などの差異が多い）もあり，配置(layout)をもって設計意思が表現され，建築上の問題（軀体，壁，床構造，天井など）や設備上の問題（電気系統，配管方式，排気配管，空調，排水など），そして飲食施設の運営上の問題に常に左右されるために生じる（原則的に不可避）。

このような厨房設計の特質から，顧客に対して常に満足のゆくサービスを提供していくには，以下のようなスタッフの欲求を確保することが求められた。

①コンサルに必要な裏付けデータ
②スタッフにも手伝ってもらえる意思伝達（対話）手段
③図面処理に付随する作業の合理化手段

つまり，職人的慣習の仕事形態から脱却していくための合理化コンセプトとその試みが求められ，設計・製図はもちろん使用機器の構成表および関連の積算業務まで，かつどの業務から入っても一元的に管理運営できることが必要となった。そこで一年以上もかけて構想をまとめ，通常のCADシステムとデータベースを専用アプリケーションで一体化することによってできる，社内呼称「CAROL―Computer Assisted drafting and information Regeneration system On Layout design（レイアウト設計における図面処理と情報再生のコンピュータ支援システム）」というコンセプトを生み出し，開発に移行した。なお，具体的な目標として以下の5点をあげた。

*かねこ　こういち　　**よりぐち　ふみお　　***いしまつ　のぼる
サンヨームラコーインターナショナル㈱　設計コンサルティング部　〒113　東京都文京区本郷3-10-15

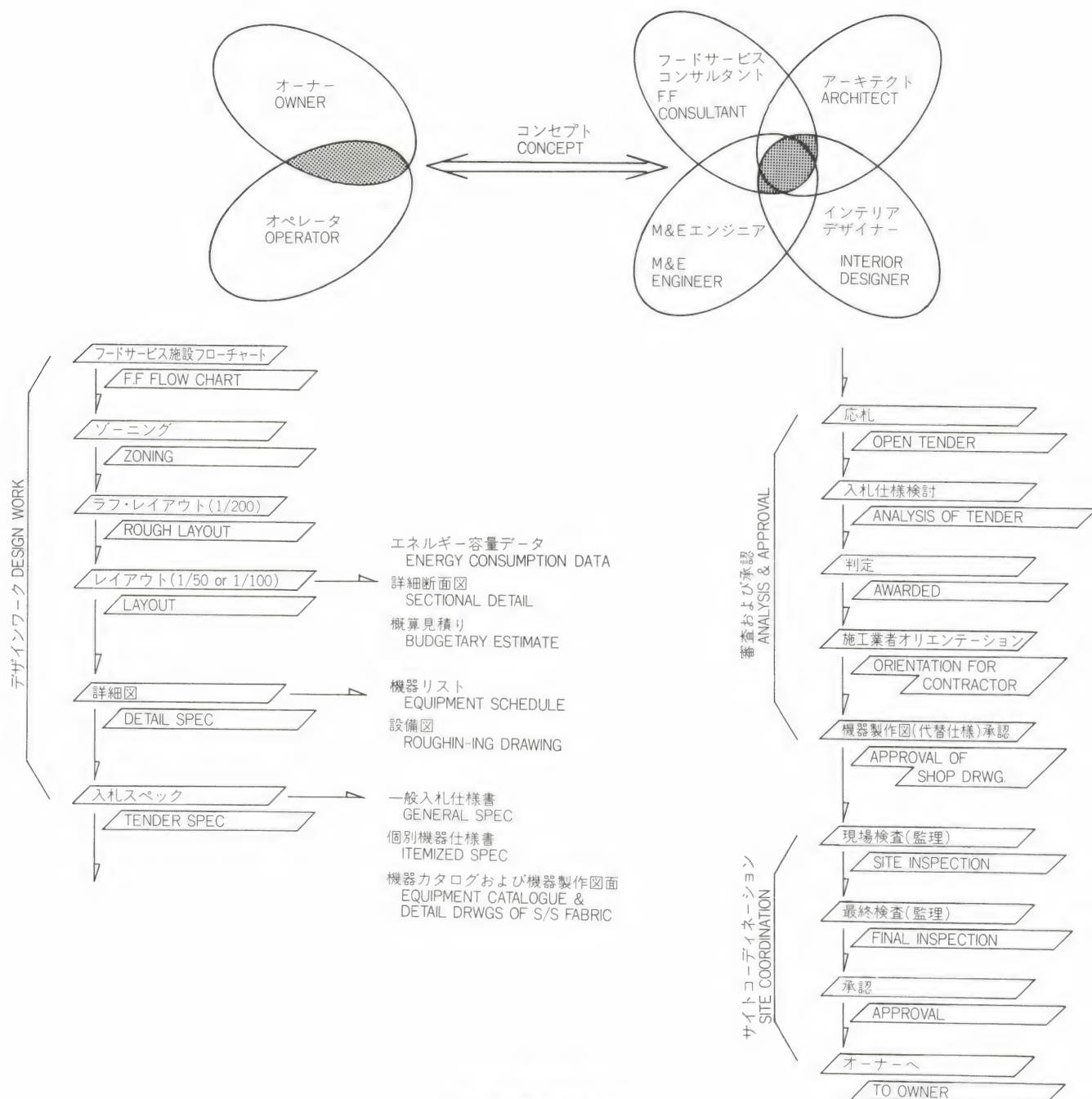



図1　厨房設計業務フロー

◉図2は次ページ。

1)　図面処理の工数削減
2)　設計変更への迅速な対応
3)　設計の効率化・標準化
4)　ノウハウの蓄積
5)　社内OA化への支援

ハードウエアおよびソフトウエア構成

図2に示すようなハードウエアおよびソフトウエア構成であるが，ハードウエアに関しては32ビット汎用型EWSが1台，グラフィック・ディスプレイが3台（スタイラスペン対応，稼働後1台追加），日本語コンソール・ディスプレイが2台，そして出力装置としてペン・プロッタ（シャープ芯対応，A0判），ハードコピー機(A4判普通紙対応モノクロ)，レーザービーム・プロッタ（400 dpi・A3判普通紙対応，稼働後追加），ドット・プリンタを各1台使用している。

特に，出力装置は図面などの目的用途別に使用機器を変え，活用している。ファクシミリ・ベースの通信および図面管理にハードコピー機（グラフィック画面の瞬時コピー機能）ならびにレーザービーム・プロッタは有効である。

ソフトウエアでは，CADシステム(ユニシス社Unidraft)から，リレーショナル・データベース・システム（RDB）のオラクル（ORACLE）へのデータリンクをもたせ，UNIXベースの厨房設計専用開発アプリケーションを仲介させた。

ハードウエア構成

グラフィック・ディスプレイ
グラフィック・ディスプレイ
グラフィック・ディスプレイ
ハードコピー
漢字ディスプレイ
漢字ディスプレイ
SS-5
固定ディスク
340Mバイト
フロッピ
A0 ペン・プロッタ
レーザービーム・プロッタ
カセットMT
漢字プリンタ

ソフトウエア構成

CAROL システム
KIT(厨房)
KMST
(マスタ管理)
リレーショナル・データベース
ORACLE
Unidraft
UNIX

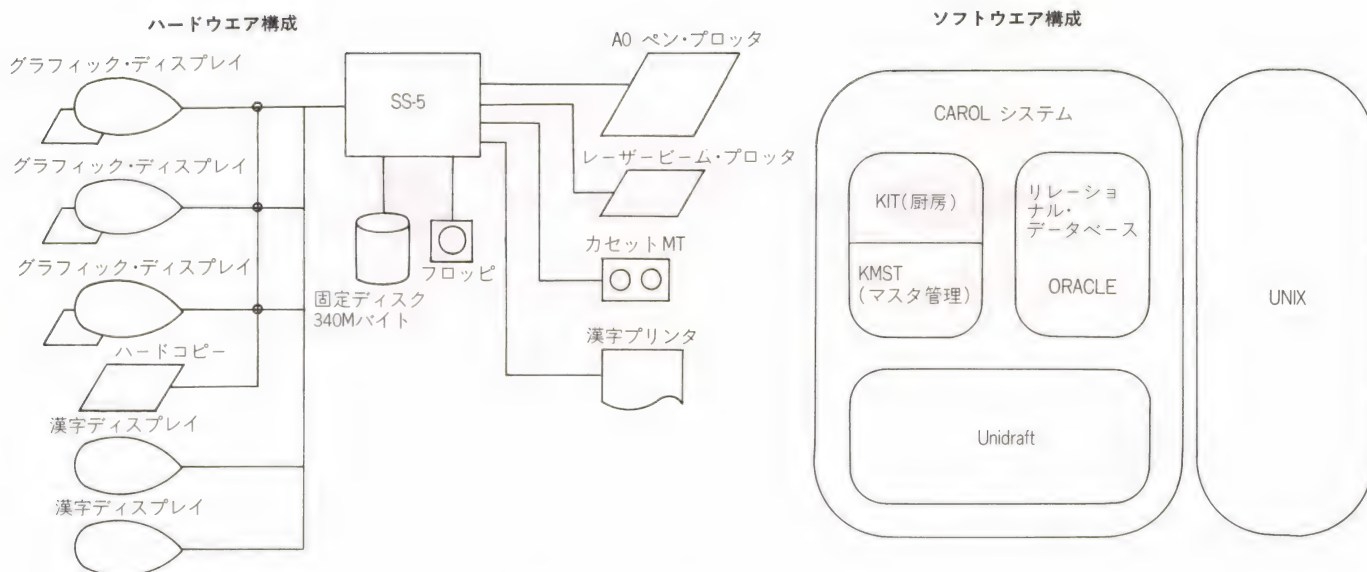



図2 ハードウエアおよびソフトウエア構成

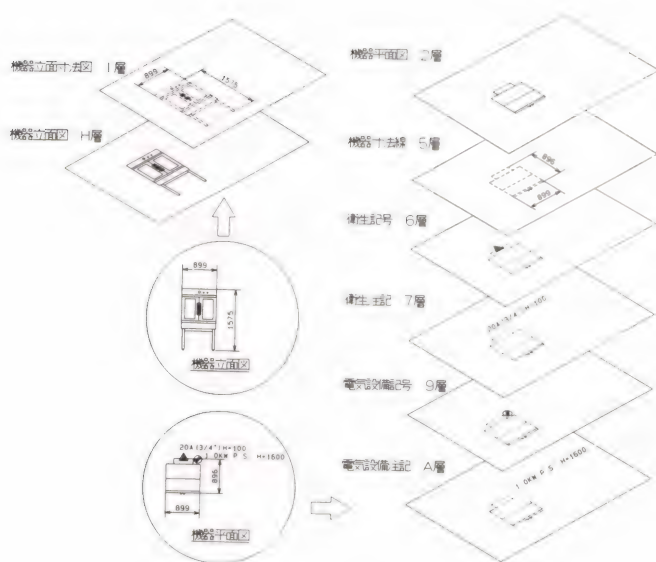



図3 機器平面・立面の図形データ登録

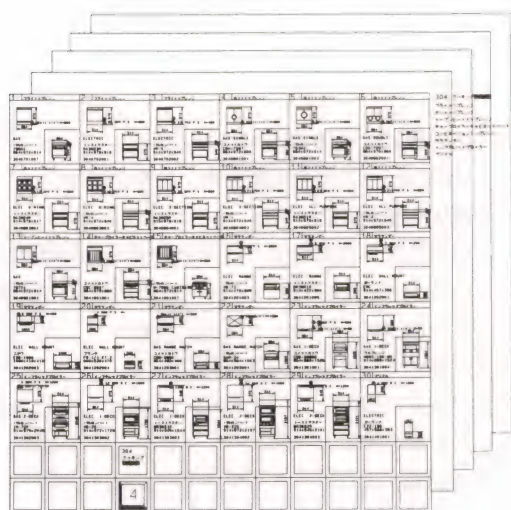



図4 メニューシート

利用面からみた機能とその効果

厨房設計におけるCADの活用を,図形処理を扱うCAD利用と,関連属性情報の処理を扱うRDB利用の2つの側面から眺めてみたい。

■ CADの利用

(1) 厨房機器図形データの標準化・一元化～ユニット図形とメニューシート

膨大な厨房機器の図形および属性データを体系立てて管理運用するため,10桁の数値コードを用いて分類した。この10桁の機器コードは,おのおのの先頭から3—2—2—3桁で階層的に名称をもつ構造とした。例えば,先頭3桁“302”で冷凍用機器,次の2桁“30207”で冷蔵庫,次の2桁“3020701”で2ドア薄型,最後の3桁で機器がモデル型番レベルを特定するといった具合である。なお,機器の名称レベルの分類はステンレス特注製品も含めて400件程度とした。

この機器コードをもとに,図形データ(機器のシンボル図表記の意味をもつ)を,**図3**に示すような平面ユニット図形として機器外形も含めてシンボル化し,使用層(レイヤー)を分け,電気／ガス／給排水／衛生などの設備記号ならびに取付け仕様(高さ,配管径)などの文字情報を注記として登録した。

また,厨房機器の立面ユニット図形も平面図形の10桁コードに拡張子を付けて同様に登録し,これらユニット図形を適宜使用する仕掛けとして**図4**に示すようなメニューシートとよばれる紙出力の検索手段を採用した。なお,現在の登録ユニット図形は平面—立面1ペアとして1,000件を,メニューシートで40枚を超えている。

(2) レイアウト図面作成～メニューシート利用のユニット図形呼出しおよび処理のプログラム化

ゼネコン，建築設計事務所からの図面をもとに建築軀体の入力を行い，CADのデータとして登録する。この際，通り芯，柱，壁などの入力は基本描画機能を組み合わせてプログラム化し，より簡単に，スピーディに行っている（軀体の自動入力装置は価格が高く，頻度も少ないこともあり，未採用）。

その軀体に沿って厨房機器を配置していくが，メニューシートを利用してスタイラスペンでヒットしたり，グラフィック画面自体で検索･呼出しを行い前述したユニット図形を呼び出し，それを回転･移動･複写しながら最適な所へ配置する。最後に，機器のID番号（アイテム番号）を，プログラム化された専用コマンドの使用によりバルーン記号を振り，**図5**に示すようなレイアウト図面を作成する。

これらの登録ユニット図形を利用すればするほど，これまで随時個々に作成していた時間が省かれる。また，ホテルやレストランにおける厨房機器（1,000種を超える場合も多々ある）のうち約40％は現在登録しているユニット図形で代用でき，約半数を占める板金物（ステンレス製品）も登録ユニットのパラメトリック処理で対応している。

(3)　立面プラン図作成～平面レイアウト図連動による自動化

平面レイアウトが仕上がっていれば，専用コマンドの利用により立面プランへ自動展開したいエリアの平面レイアウト図形を連続ヒットし，次に立面プランを表現したい開始点をヒットするだけである。この操作を表現したいエリアごとに行い，立面プラン図を作成する。また，平面レイアウト図が仕上がっていなくても，メニューシート利用により平面レイアウト図同様に扱える。立面プラン図例を**図6**に示す。

これまで手書きでは修正・訂正に手間どるので，提出対応がほとんど最終プラン段階のみであったことに比べ，説得力のある対応が初期段階より可能となった。

(4)　設備図作成（電気／衛生）～層の有効活用と自動化

厨房設備機器の配置に伴いレイアウト図面が作成されると，次に関連設備への接続サイズと位置，エネルギーなどの消費量をはじめ，配管工事や配線工事などの厨房関連工事上必要な事項を記述した設備図面の作成に入る。

この際，平面ユニット図形の使用層区分に従い表示層を選択すると，電気設備図（**図7**），衛生（ガス設備含む）設備図となる。レイアウト図が仕上がっている場合はほぼ設備図が出来上がった状態となり，軀体との関連を再考して修正・訂正する。

また，機器配置はさまざまな角度で行われるので注記などの文字列も同様にさまざまな角度をもつが，専用コマンドを利用して初期化（横一列水平に並ぶ）と任意位置への移動を行う。

層の使い分けは，例えば軀体は0層，軀体寸法は1層に，厨房機器の平面は2層にといったように行われ，こうすることにより0層と1，2，3層を出力対象にすればレイアウト図面となる。0，1，2層と機器の電気記号を登録している9層，電気容量および接続位置が記述登録されているA層をともに出力すると電気配線工事用の設備図（電気）となる。給排水やガスなどの配管工事用の設備図（衛生）も同様の層操作で出力できる。なお，使用層の区分は紙出力の際にシャープ芯，筆圧および線の太さの選択とマッチングさせることができ，いわゆる濃淡の

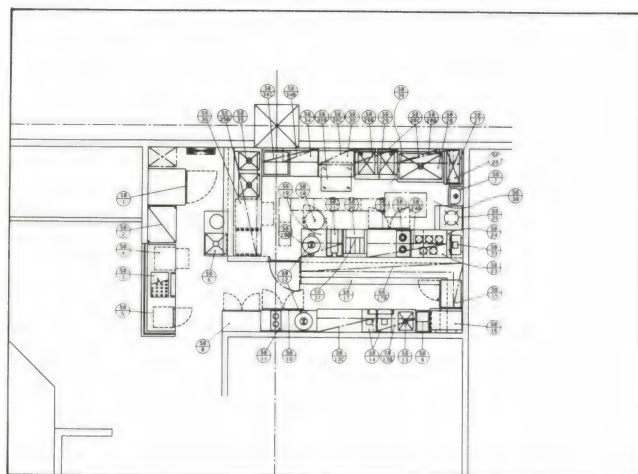



図5　平面レイアウト図

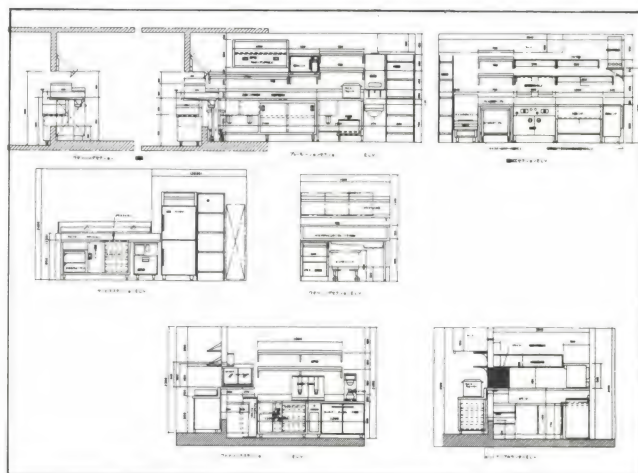



図6　立面プラン図

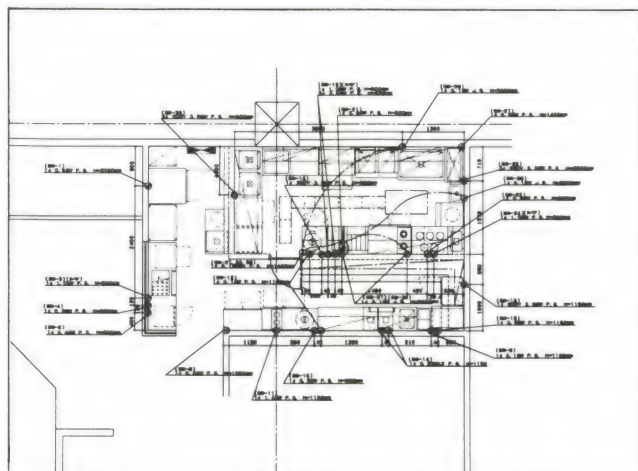



図7　設備図（電気）

図8 KMSTメニュー画面

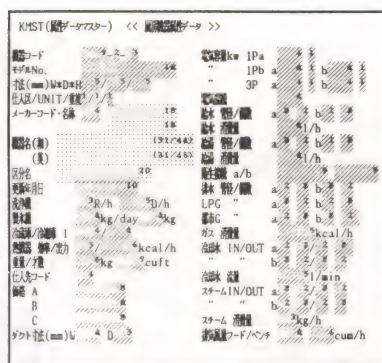



図9 KMST属性データ管理画面

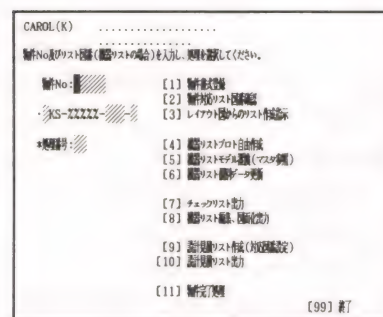



図10 KITメニュー画面

線画表現を実現している。

■ RDBの利用

(1)厨房機器属性データの標準化・一元化～ORACLE利用のデータベース化

従来，膨大なカタログ情報の中から探し出していた厨房機器の仕様情報(外寸，電気容量，ガス容量，各種管径，価格など)も標準化を行い，RDBシステムのORACLEを用いて機器属性データのデータベースとして一元化した。また，その属性データの管理においてはKMST(厨房機器属性データマスタ管理システム）としてプログラム化し，誰でも簡単に検索でき，データ・メンテナンスおよび拡張にも利用できるようにした。**図8**にKMSTのメニュー画面を示す。

厨房機器の属性データでは，機器コード10桁の定義によって**図9**に示すように以下の項目を管理している。

1) 仕様項目：モデル型番，外形寸法，電気・ガス・給排水容量および管径など
2) 性能項目：洗浄能力，製氷能力，保管容量など
3) 見積項目：価格，才数，重量など
4) 機器名称：機器コードの先頭5桁で設定
5) 区分名称：機器コードの先頭7(5—2)桁で設定(機器を性能区分する項目で，2ドア冷蔵／4ドア冷凍／4ドア冷凍冷蔵など)

さらに，付帯管理項目として次の項目もコード設定にて同時に管理している。

6) メーカー名称／仕入先名称
7) 仕入先区分／図形ユニットの有無／モデル型番重複
8) 仕向地名称（飲食施設の立地地域区分)

このデータベース化によって必要な項目(例えば，幅800，2ドアの冷凍・冷蔵庫や，ある特定メーカーのすべての機器などの複数要求設定も可)をキーとして検索し，機器の仕様・性能・価格などを自在に調べることが実現できた。現在，属性データの登録件数は国内・海外の機器，板金製品を含めて3,000件を超えている。

(2) 機器リスト作成～平面レイアウト図連動の自動化

平面レイアウト図作成と同時に配置した機器の設備情報（属性データの一部)をリスト化する作業について，CADシステムとRDBシステムをリンクして自動化を図った。また，機器リストの作成プロセスを前述の属性データのデータベースを利用するかたちで，KIT(厨房機器リスト化および情報再生システム)としてプログラム化し，レイアウト図が仕上がっていなくても，かつどこから仕事を開始してもよいようにした。**図10**にKITのメニュー画面を示す。

まず，漢字コンソール画面でこのKIT（厨房機器リスト化および情報再生システム）を実行し，物件の名称・受注形態・英文／和文出力の選定，仕向地・電源仕様・ガス仕様などの地域特性情報，または機器リスト作成にあたっての配管径単位（A呼称，インチ，ミリ)，寸法単位(ミリ，フィート）などを登録する。これを物件書式登録とよんでいる（作業領域を確保し，同時に図面作成から終了までの一連の作業で作成する図面，リストファイルを管理制御する)。

次に，レイアウト図面から一気に機器リスト構成を作成するため，リスト作成指示をかける(配置したユニット図形の機器コードとアイテム番号を抽出し，属性データベースとリンクさせてRDB上でリストファイルを作る)。これを図面化するにあたり，A1和文ヘッダー，A1英文ヘッダー，A2和文ヘッダーの3種類や，ライン・スキップ数および層分けなどを選択すると，先に設定した物件書式にそって自動編集・図面化される。この出力例を**図11**に示す。

レイアウト図面を作成していなくても，機器リストプロト自由作成という処理により，アイテム番号，機器コードもしくはモデル番号，員数，属性データ参照有無を定義するだけで，レイアウト図がある状態と全く同様に扱うことができる（独自にリストファイルを作り，それをRDB機能を用いて変更・訂正・追加および属性データ検索を行いながら編集する)。

そのリストファイルを使用用途（設備情報，設計情報，見積情報）に応じてチェックリストとしてプリンタ出力し，図面化しなくても結果を把握し，赤字チェックを行って誰にでも作業依頼ができるようにしている。

また，個々の機器リストの集合で一つの物件が構成されるので，その対応を定義し，物件全体の設計見積も自動的にとれるようにしている。特に，この見積情報はパソコン・ファイルとしてフロッピディスクに変換される機能も取り込み，社内OAシステムへ連動できるようにしている。仕事の完了時には，その物件（飲食施設）全体の特性を分析した帳票も作れるように

CAD活用上の課題と展望

■厨房業界および関連業界におけるデータ流通の課題

飲食施設の厨房設計におけるCAD活用には，厨房機器ユニット図形とその属性データのデータベース化が一義的には最重要で，このことなくしてCAD化による効果は得られないと考える。また，このような試みが必要とされるところに厨房業界および関連業界におけるCAD利用の壁があり，データ作成とそのメンテナンスは大きな課題となる。

アメリカでは，すでに厨房機器メーカーから設備機器の仕様，図形データ類が直接フロッピディスクにより無料提供されており，日本でもこのような試みの展開が待たれる。

■ CAD/RDBの相互利用の展開

飲食施設の機能設計では，オーナー・オペレータの要望を，ゼネコン・建築・関連工事業者との打合せを介しながら，設計者のノウハウをもとに具現化し，最適な施設機能を提案していく。このノウハウ（機器の特性・性能の把握および必要能力の算定／施設における食材・人の流れ／飲食環境の動向など）を設計者の個々の知識・経験にとどめずデータベース化し，数値化および汎用化する必要がある。

その意味で，RDBによる情報の蓄積・分析・統計化とその応用におけるAIの活用を模索しているところである。同時に，その施設機能をよりわかりやすいかたちで提案するため，**図12**に示すようにCADの機能を用いたよりビジュアルなシミュレーション（パース図，アイソメ図など）へ展開させたいと考えている。

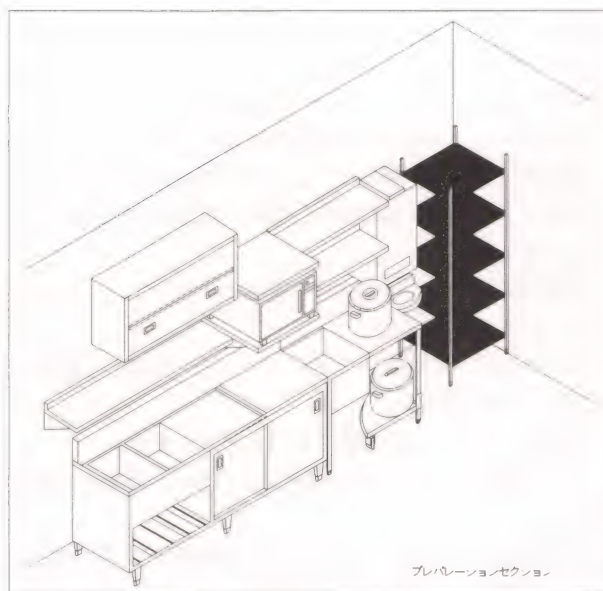



図12　アイソメトリック

(103ページより続く。)

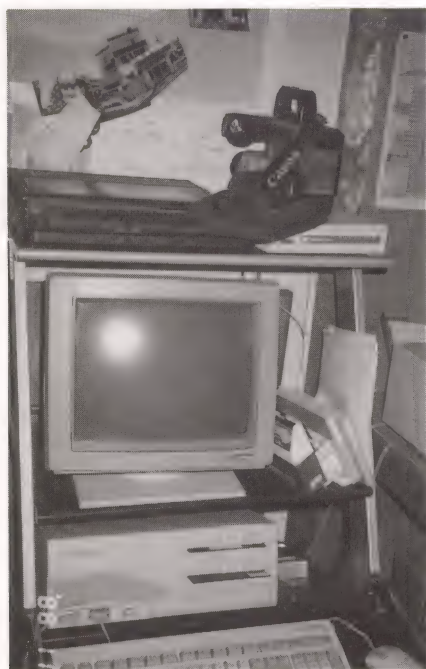



図19　アークス建築研究所（所内には外部とのデータ通信用の端末を1台設置。）

おわりに

以上でみてきたものは専業の建築設計事務所が中心となっているが，その他にもゼネコンの設計部や大学の研究室などさまざまなかたちでCADを利用した建築の設計を行う組織がある。前者は企画・設計から施工・メンテナンスにいたる総合的なCADの利用が可能であるし，後者は設備投資効率にとらわれない新鮮な発想によるCADシステムの提案が可能である。また海外に目を向けてみるとCAD一辺倒の設計事務所はほとんどなく，従来の設計の枠組みに調和させながらCAD化しているケースが多いときく。もちろん，日本を取り巻く社会環境は欧米と異なり，もはや参照すべき範例を他に求めるのではなく，独自の立場に立ってCADと設計業務のかかわり方を模索しなければならないとすれば，新しいタイプのCADコンサル型の設計事務所を中心に設計事務所の再編成の時期が遠からず到来すると思われる。その段階では，設計事務所のOA化も単なるOA機器の導入・普及の段階から，設計者自身がハードウエアとソフトウエアを快適な環境の中で自由に使いこなすという夢（？）のような段階へ移行し，人間とコンピュータとの自由で創造に満ちあふれた新しい関係が確立される時代が実現することになるだろう。

参考文献

1）北代禮一郎，中山信二：「設計事務所のOA化を目指して」，新建築，1988.6

2）中山信二：「設計事務所の総合的OA化」，建築知識，1988.9

3）CAD研究会：「実用段階に入ったパソコンCAD(1)～(3)」，建築文化，1988.7,9,11

4）笹田：「都市計画・建築デザインとCG/CAD」，NICOGRAPH'88

5）山口重之：「建築と環境のデザインとコンピュータの接点 領域で」，PIXEL，No.68，pp.100-104，1988.5

シミュレーション・ツールとしてのゴルフ場造成設計システム

浅沼組では，ゴルフ場造成設計を支援するビジュアル・シミュレーション・ツールを独自に開発した。このシステムの構成から実際の活用までを紹介する。

伊藤　英和*

はじめに

当社の土木分野における設計業務のシステム化は，1981年に開発した「造成実施設計システム」が最初であり，これはホスト・コンピュータによるバッチ処理であった。1985年に図面の作成効率および品質の向上と設計の標準化を推進するためCADを導入した。それらの効果を上げるため，「擁壁配筋図自動作成システム」「構造図自動作成システム」を開発し，さらに顧客のニーズ（より速く，ビジュアルに，高品質な）にこたえるべく，構想企画・基本設計段階でのシミュレーション・ツールおよびプレゼンテーション・ツールとして，「土地造成設計支援システム」「ゴルフ場造成設計支援システム」を開発した。

システムの構成

ゴルフ場造成設計支援システム（ACE-GP）は以下の構成となっている。

■ハードウエア

- ●ホスト・コンピュータ　FACOM M760/8
- ●グラフィック・ディスプレイ　F6240A
- ●プロッタ　GP5400
- ●デジタイザ　DH-7000

■ソフトウエア

1)　地形入力サブシステム
　パーソナルコンピュータに接続されたデジタイザ入力
2)　地盤高メッシュ変換サブシステム
　ホスト・コンピュータによるバッチ処理
3)　ゴルフ場設計支援サブシステム
　ICAD/SDS3上で構築した対話処理
4)　図面作成・出力サブシステム
　ホスト・コンピュータによるバッチ処理

サブシステムの概要

■地形入力サブシステム

等高線・境界点・進入路などの地形データを作成し，ホスト・コンピュータに転送するものである。

■メッシュ変換サブシステム

等高線データを格子点上の高さデータに変換し，さらに等高線・境界点・進入路などを現況平面図として作成するものである。これらはホスト・コンピュータでバッチ処理される。

■設計支援サブシステム

現況平面図上に対話シミュレーションを重ねながら計画を進めるものである。この作業を支援するために，現在62種の業務コマンドで構成し，条件設定～設計作業～計画案保存までの一連の設計支援を行っている。それらの主な機能として以下のようなものがある。

1)　コース・レイアウトの計画進捗レベルに応じて，以下の3つの設定法が選択できる。

- ●マニュアル設定法：設計者が各ホールの位置・高さを設定する。
- ●半自動設定法：設計者は各ホールの位置のみを設定し，高さ

*いとう　ひでかず　㈱浅沼組　電算室　〒543-91　大阪市天王寺区東高津町12-6

図１ 造成計画平面図

はシステムが自動設定する。

● 全自動設定法：各ホールの位置・高さともにシステムが自動設定する。

2) 近似直線で入力した計画線は，設計者のデザインイメージに近い自由曲線で表示する。

3) 計画のチェックをビジュアルにするため，任意の断面やホール縦断面の表示，法面のすりつき位置自動発生などを行っている。

4) 土量・面積他の数量算出などをリアルタイムに表示する。

5) 計画の変更は，たとえ局部的であっても関連データに広く影響を及ぼすため(例 プレー線変更：グリーン，ティー，フェアウェイなどの図形データ変更→法面すりつきデータ変更→面積データ……)，データの一元管理によって自動的に整合させている。

■図面作成・出力サブシステム

1) 造成計画平面図（**図１**）

造成計画のもととなるもので，現況平面図上に計画線（各ホール・進入路・クラブハウス・池・残存緑地など），ヤーデージ表などを描いたものである。

これは設計支援・図面作成などの業務コマンドにより，対話

表１ 土量計算書

土 量 計 算 書

件 名 NO. 1

測点番号	工期 距離 (m)	切土 断面積 (㎡)	切土 平均断面積 (㎡)	切土 土量 (㎡)	盛土 断面積 (㎡)	盛土 平均断面積 (㎡)	盛土 土量 (㎡)
0	0. 0						
1	3 0. 0	119 90	59 95	1,798 50	356 80	174 40	5,352 0
2	3 0. 0	586 70	353 30	10,599 0	1,335 80	846 30	25,389 0
3	3 0. 0	1,515 60	1,051 15	31,534 50	1,568 0	1,451 90	43,557 0
4	3 0. 0	4,072 70	3,551 95	106,558 50	2,641 40	2,104 70	63,141 0
5	3 0. 0	8,231 20	6,151 95	184,558 50	1,505 50	2,073 45	62,203 50
6	3 0. 0	7,415 80	7,823 50	234,705 0	1,574 10	1,539 80	46,194 0
7	3 0. 0	6,317 50	6,866 65	205,999 50	4,599 0	3,086 55	92,596 50
8	3 0. 0	9,688 10	8,002 80	240,084 0	3,047 20	3,823 10	114,693 0
9	3 0. 0	7,935 20	8,811 65	264,349 50	2,720 50	2,883 85	86,515 50
1 0	3 0. 0	9,006 10	8,470 65	254,119 50	3,743 30	3,231 90	96,957 0
1 1	3 0. 0	6,797 30	7,901 70	237,051 0	6,676 80	5,210 05	156,301 50
[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]

図２ ホール設計書（平面図・縦断面図）

図３ ホール設計書（横断面図）

●図2，3，表1は前ページ。

型式で作成される。

2) ホール設計書（**図2，3**）

各ホールごとの平面図・縦断面図・横断面図を描いたものであり，これは全体の計画・地形データから各ホールごとに取り出して図化している。平面図は自動的にレイアウトされ，縦断面図は平面と照合できるようプレー線の高さと現況高さを表示している。横断面図はホール横断形状を表しており，断面位置および高さは自動的に決定され，これらはバッチ処理される。

3) 造成横断面図（**図4**）

土量計算のもととなる横断面図を描いたものである。これは対話処理時に計算範囲と断面ピッチを画面から入力し，リアルタイムに土量を表示する。この対話処理時の根拠となる断面図および土量計算書（**表1**）をバッチ処理して出力される。

4) 景観図（**図5**）

任意の視点と注視点を画面で指示し，3次元処理したものである。

システムの構築方法

■データの縮小化

一物件に複数の計画案の発生することが多いが，現況地形データは唯一であるので，各計画案の共通データとして独立して保存している。

■2次元処理

対話処理時のデータの取扱いは，データ領域・操作性・レスポンスなどを考慮して2次元として扱っている。平面と断面とのデータ相関はアプリケーション上で整合させている。

■バッチ処理

提出設計図書として完成させる図面（造成横断面図・ホール設計書の作成など）は処理の効率上，バッチ処理で行っている。

■設計の支援処理

設計の支援処理部分は，開発やレベルアップ時の効率を考慮して業務コマンド単位で独立させている。

システムの活用例

設計には，大きく分けて基本構想，基本計画，実施設計という3つのレベルがある。この基本構想・基本計画の段階において，「シミュレーション技法を駆使して設計を進めたい」「ビジュアルなプレゼンテーションをしたい」「設計の効率化・高速化を図りたい」「早期に計画の成否を判断したい」などの欲求がある。これらの背景には近年の第3次ゴルフブームによる設計依頼の物件数が日ごとに増加しているといった状況がある。これらの計画物件を無駄なく消化し，効率を上げつつ高品質な設計を提供するということがシステム構築の基本コンセプトとなっている。

当システムの利用手順は，以下のように行っている。まず地形図より地形データを作成する。その現況地形図をもとに画面との対話によるシミュレーションを繰り返しながら計画を進める。さらにこの計画案に肉付けをして，図面作成や数量計算などを行い，提出図書としてまとめる。また，コースや周辺地形を含めた景観図を加えてビジュアルに表現する。

現在の運用体制は，専任オペレータにより処理を行っている。オペレータは設計者が描いたラフプランをもとに計画を入力し，法面すりつき状況や土量バランスなどの数量を検証し，設計者とのコミュニケーションを図りながら計画を進めていく形態をとっている。オペレータ専任については是非もあるが，現状では設計者への教育と操作習熟が困難なことからこの運用がベストであると考えている。現状では，1件（18ホール・クラブハウス用地などの設計業務）当たり4日間で造成計画平面図，土量計算書，ホール設計書，景観図などを一式書類として仕上げている。

また，すでに設計された物件においては，計画案を直接デジタイザから入力し，提出図書を作成する機能もある。

当システムの主な利点は，対話型シミュレーション機能によりコースレイアウト検討の段階で高さ決定を短時間で行えることにある。計画（高さ・位置）を設定する際，設計者のプランをそのまま入力する「マニュアル法」，平面プランだけを入力し，システムがその地形データをもとに前後ホールとのかかわりを考慮しながら9ホール単位で最小土量が得られる計画高を設定する「半自動法」，さらに9ホール分の計画範囲の入力だけでシステムが各ホールのレイアウトの設定から高さの設定までをする「全自動法」が状況により自在に選択できる。現在では，熟練設計者の意図を反映できるという点からマニュアル法を使い，土量バランス・法面すりつき状況・その他数量を確認しつつシミュレーションを重ねて計画を煮つめるという活用例が多い。しかし今後，若手設計者や急を要する立案業務の増大が予想され，半自動・全自動法によるシステムからの提案をもとに設計を進めていくケースが多くなると考えている。

景観図については，設計上の整合性や設計者のデザインイメージの確認の他に事業主への企画提案には欠かせないものとなっており，各ホールごとの平面・縦断・横断図をホール設計書として出力し，造成計画平面図，土量計算書（各造成横断図）などを加えて事業主に提供している。

おわりに

当システムの開発にあたっては，ゴルフ場設計に熟練した設計者の意見を十分に吸い上げ，分析を重ねたが，今後，時代の変遷と顧客のニーズの変化に対応して機能の拡大，拡充を図っていくことが必要だと考えている。

現状における課題としては以下の通りである。

1) 地形入力において労力・時間（3～5 h）がネックになりつつあるが，今後はオートデジタイザの利用を図りたい。
2) さらに完成度の高いビジュアルなプレゼンテーション・ツールとして，CG画像のサーフィスモデル作成によるレンダリング，動画の作成などを図りたい。
3) 当システムの利用者の拡大（専任オペレータから設計者自身によるオペレーション）を図るべく，オペレーションをさらにわかりやすく簡便に使用できるように改善，改訂を加えたい。

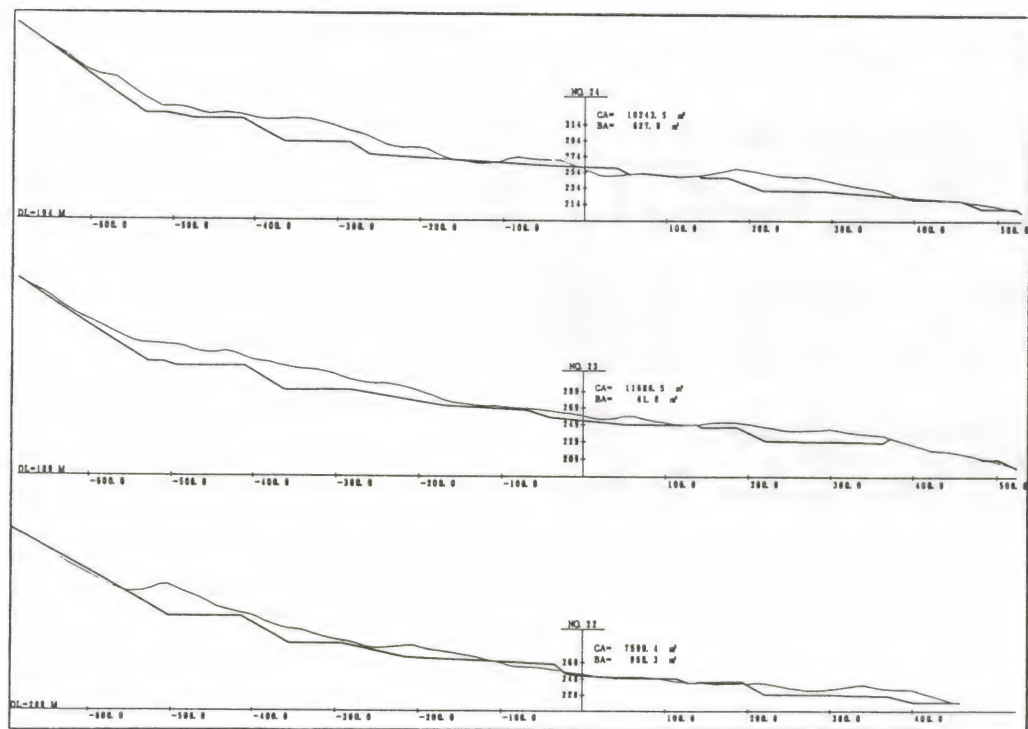



図4 造成横断面図

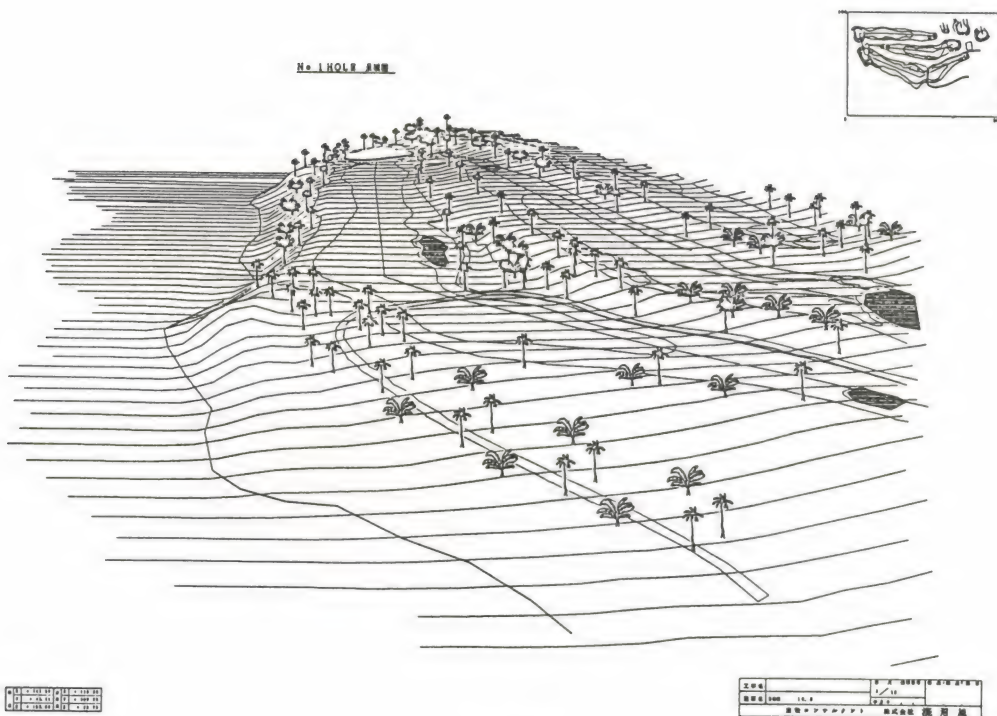



図5 景観図

特集２　建築・土木のCADとCG

大成建設におけるCAEの一例 自然力に対する構造物の応答性状をさぐる

自然力に対する構造物の応答性状を把握することにより，構造物の合理的設計ならびに快適な居住空間の創造を目指す研究と，CAEのかかわりの一端を紹介する。

山田　正明*

実現象を知る

構造物の設計にあたっては，自然から受けるさまざまな力，例えば地震や風に対して構造物がどのような動きをするのかを把握するとともに，十分な安全性を保有していることを確認する必要がある。また，これらの性状を把握することにより，自然環境に対して快適な居住空間を創造するための資料を得ることができる。

わが国では，高度な耐震，耐風設計法が普及するとともに建築基準法などの法令が整い，世界的にもレベルの高い構造物が作られている。しかし過去において，いくつかの大地震などにより予想外の被害を受けたことも事実である。これらの被害の教訓は学問の進歩を促すとともに新たな知見として蓄えられ，設計者や研究者にとって大きな財産となってきた。

新たな構造物を建設するにあたり，必要とする技術がそれまでに蓄えられた知見の範囲内あるいはその延長線上にある場合には技術的困難を伴うことは少ない。しかし，これまでに経験のない構造物，例えば既存の超高層ビルをはるかにしのぐ超々高層ビルを建設するといった場合に解明しなければならない問題点がいくつかある。その一つが先にあげた自然力に対する構造物の挙動の把握であり，また構造的安全性の確認である。

これらの目的のために実作業として実験や解析を行うわけであるが，現象が複雑な場合や対象物が大きい場合には得られるデータ量が膨大になり，個々のデジタル値だけをみても現象の全体的把握は難しい。したがって，画像による現象の把握などが必要となる。

実験と解析

実現象を知りそれを実構造物に適用するにあたって，実験と解析とは車の両輪のような存在であり，どちらか一方だけで十分ということは少ない。

まず，実験についてみると，実構造物を用いて応答性状一般を把握することは困難である。

この主な理由は，実構造物では自然力および構造物の特性をパラメータとするパラメトリック・スタディを行うことが困難な点にある。したがって，模型を用いた実験を行い，模型則を介して実現象を把握しなければならない。しかし，模型則を適用しづらいパラメータもあり，実験結果の数値そのものをもって即座に実構造物に適用することができない場合も多い。

一方，解析では過去の知見に基づく種々の仮定を用いる。したがって，既存の知見の範囲内にある現象について検討するときはよいが，その範囲外の問題を取り扱う場合には採用した仮定が本当に正しいかどうか吟味する必要がある。この場合，実験によって実現象を把握し，その現象を正しくシミュレートで

図１　三軸振動台

*やまだ　まさあき　　大成建設㈱　技術研究所構造研究室　〒245　神奈川県横浜市戸塚区名瀬町 344-1

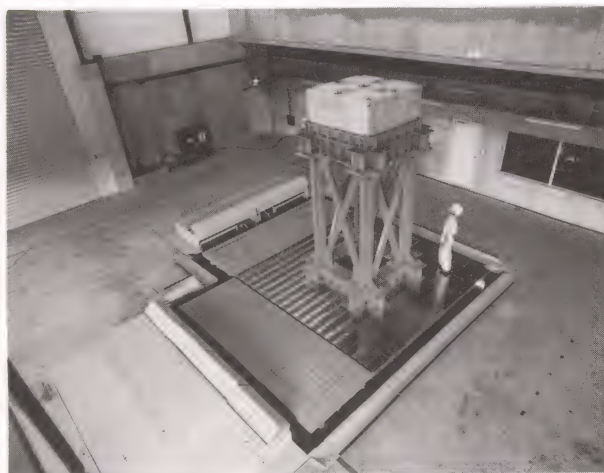
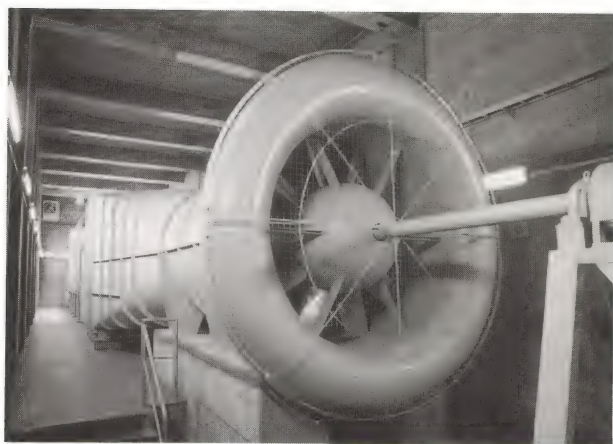



図2 風洞送風機部分

図3 風洞風下側

きる解析コードを開発し，そのコードを用いて実構造物の挙動を推定したり，安全性を確認したりするという手順を踏むことが多い。

実現象を把握するための実験が模型実験となる理由は先に述べたが，その理由となるパラメトリック・スタディのしづらさは実構造物の大きさと実験施設の規模との関係に一因がある。地震力を想定した実験では振動台を使用することが多いが，振動台の大きさや加振力には制限がある。したがって，実構造物を用いるのではなく，振動台の制限に合わせた模型試験体を製作し，その試験体を用いて実験を行わなければならない。大成建設では**図1**に示す大規模な三軸振動台（水平2方向および上下方向に同時加振が可能）を所有している。しかし，この大型振動台でも平面的な大きさは4 m角，最大載積重量は20 tであり，実構造物をそのまま搭載するわけにはいかない。さらに構造物と地盤との動的相互作用を考慮した実験ではこの振動台上に模型地盤を作り，その上に構造物模型を設置する。このような模型試験体を用いた実験では，模型則を介して実現象の把握を行うが，一般的には実構造物と模型との長さ比を定め，この長さ比を基準として各パラメータ，例えば重量，時間軸，変位，速度，加速度などを実構造物と模型とで対応付ける。しかし，減衰のように模型則の当てはめづらいパラメータもある。

また，これらの模型は実構造物と形状・外観ではなく，振動性状を一致させる必要がある。それゆえ，外観のイメージは模型と実構造物で異なることが多い。

一方，風を想定した実験では自然風を模した風を人工的に発生させ，この風を利用して各種実験を行う風洞施設を使用する。大成建設では測定部の幅が2.0 m，高さが1.5 mの大型風洞を所有している（**図2，3**）。しかし，振動台実験と同じようにこの風洞内に実構造物を設置することはできず，模型を用いる。構造物が風によって力を受ける場合，その形状によって受ける力に大きな差が現れる。したがって，風洞実験に使用する模型では形状まで実構造物と一致させる必要がある（**図4**）。

画像で応答性状を知る

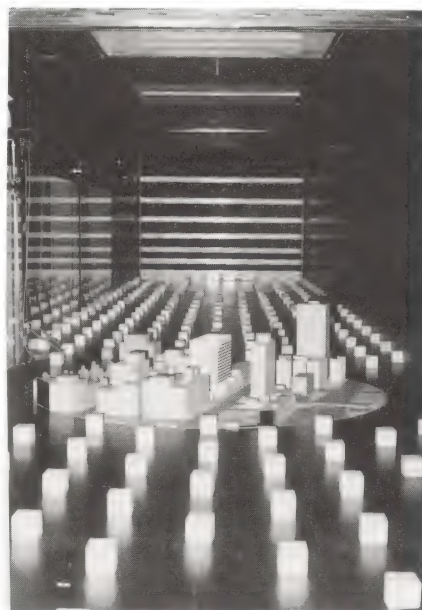
実験あるいは解析で得られる膨大なデータをデジタル値でみても現象の全体的把握は難しい。着目する現象を把握するうえで必要とするデータをいかに素早くピックアップし，いかにわかりやすく表現することができるかが実験や解析を効率的に進めるうえでの決め手となる。

構造物が揺れる状況は外力の性質によっても異なる。しかし，その表現方法は地震力による場合と風力による場合とでは大きな差はない。また，応答変位や応答加速度を取り上げてその最大値のみに着目するのであれば，デジタル値で表現することができる。しかし，揺れ方全体を把握するには構造物の動的挙動を可視化した方が理解しやすい。さらに，ある現象に着目し，その現象と構造物の応答との関係をとらえるためには，着目した現象に関する時系列データやその可視化も必要となる。

次に，振動実験により得られるデータの一例を紹介する。

一般に，構造物の基礎は底面全体が地盤に接している。しかし，外部から非常に大きな力が作用した場合にはその一部が地

図4 風洞内部と構造物模型

◉ 図 8，9，13，14 は COLOR IMAGES 参照。

盤から剝離することも考えられる。その場合，構造物は剝離以前とは異なった応答性状を示す。**図 5** に，この性状把握のための実験に使用した振動模型の概略を示す。これは，あらかじめシリコーンゴムで製作した地盤模型を振動台上に固定し，その上に 5 質点の構造物模型を設置した例であり，試験体としてはかなり単純である。主に基礎底面と地盤の剝離状況を計測するため，基礎底面と地盤上面との間に 64 台の近接スイッチと 20 台の圧力センサーを取り付けた。また，構造物の全体的挙動を把握するため，試験体の各位置に 23 台の加速度計と 6 台の非接触型変位計を設置した。基礎底面でのセンサーの配置を**図 6** に示す。これらのセンサー類により得られるデータはすべて時系列データであり，1 回の加振だけでもかなりの量になる。これが加振方向（例えば X 方向加振，Y 方向加振，Z 方向加振）やその組合せを変えたり（X 方向と Y 方向，X 方向と Z 方向，Y 方向と Z 方向，あるいは 3 方向同時加振），入力波の性質を変えたり（国内外で得られた観測波や正弦波）しながら数百回もの加振を行うわけであるから，単純な試験体においても得られるデータ量はかなりのものである。さらに，実験の進行にあたっては前ステップの加振結果に対して工学的な判断を加え，実験が誤った方向に進んでいないことを確認したうえで次の加振を実施する場合も多い。これらの点を考えると，膨大なデータの中から必要とするデータをいかに素早くピックアップし，いかにわかりやすく表現できるかが実験などを効率的に進めるうえでの決め手となることが理解できよう。

先の模型試験体に水平 2 方向から地震力を加えた場合の代表的なセンサーの出力波形例を**図 7** に示す。(a) が作用した地震力（外力）であり，振動台上 X 方向の加速度波形，(b) が構造物模型頂部中央での X 方向応答加速度波形である。この応答加速度波形は基礎が地盤から剝離することにより，剝離をしない場合に比べてかなり異なった特徴を示している。つまり，この実験では基礎が地盤から浮き上がることにより，構造物の応答性状がそれ以前に比べてかなり変化することを示唆している。

次に，基礎が地盤から剝離した状況を示す波形の一例を(c)と(d)に示す。(c)は基礎最外端における圧力センサーの出力波形，(d)は近接スイッチの出力例である。(c)では下側の出力値が変化しなくなった（波形の頭が切れた）時刻領域，(d)では波形が非接触を示す時刻領域で剝離現象が生じていることを示している。このように，これら個々のデータよりセンサー設置点における剝離状況を知ることができる。しかし，基礎底面全体の状況を判断することは困難であり，全点のデータを集めて面としての剝離状況を表現する必要がある。近接スイッチならびに圧力センサーの全データから全底面に対する剝離面積を求め，浮上り率として表現したのが (e) である。この図により，基礎底面の剝離状況が時間とともにどのように変化するかを把握することができる。さらに，この状況は解析的にも表現することができる。例えば，**図 8** は解析的に求めた剝離状況と接地面における地反力分布の時刻断面図であり，ライトブルーで剝離面および地反力のほとんどない部分を示している。また，接地面では色の変化ならびに等高線の高さにより地反力の大きさを表現し，等高線が高いほど大きな地反力であることを示している。なお，剝離面での等高線は完全に 0 である。**図 9** は，図 8 から等高線の表現を除いて真上から連続的に見たものである。このような見方をすることにより剝離がどこから開始し，どのような角度でどのように進行するかを知ることができる。また，同

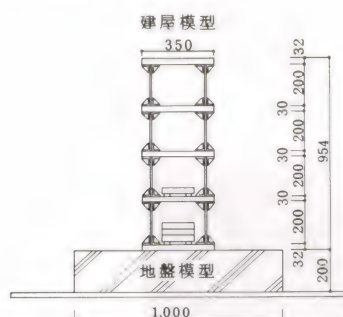



図 5 振動模型

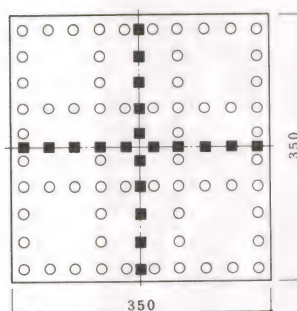



図 6 基礎底面センサー

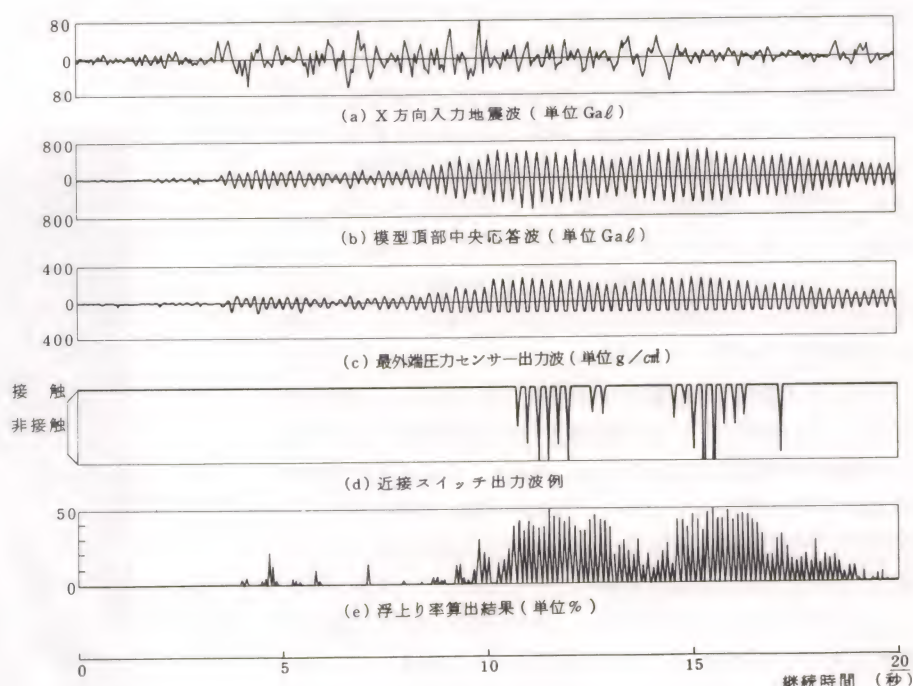



図 7 出力波形

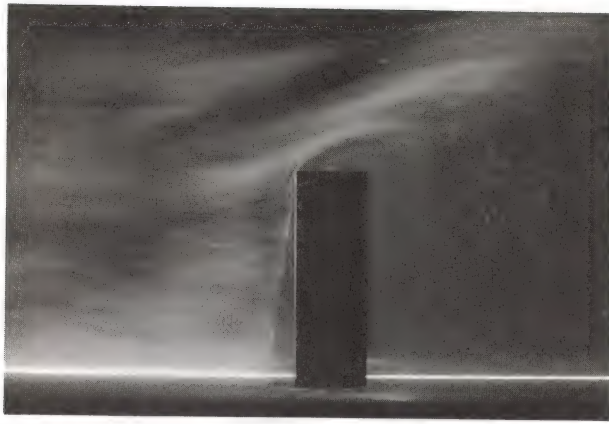



図10　空気の流れの実験(1体)

図11　空気の流れの実験(2体)

じ表現を実験結果についても適用し，両者を比較することにより解析コードの改良ならびに検証を行うことができる。このようにして検証された解析コードを用い，基礎に剝離を生じることにより現れる上部構造物の応答性状の変化を正しく評価し，実構造物へ適用することが実験ならびに解析を行ううえでの最終目標である。

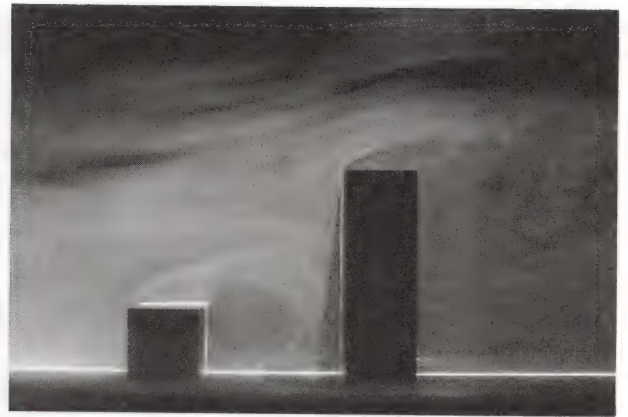
なお，上部構造物の振動解析手法あるいは解析結果の表現方法などについては種々の方法があるが，その紹介は別の機会に譲る。

風外力の把握

構造物は自然界から力を受けて振動を生じる。その応答評価をするには，まず構造物に作用する力，つまり外力を仮定しなければならない。その仮定の難しさを，地震と風を比較すると地震よりも風の方が難しい。もちろん，地震力についてもいろいろ難しい面はあるが，構造物の建設地点を決めることにより，外力そのものを直接仮定できるという点で風よりも地震の方が取り扱いやすい。

風によって生じる外力は，風の吹く方向だけではなく，風に直交する方向にも生じる。これは構造物のまわりを風が流れ，その流れに応じて発生する空気力が外力となって構造物に作用するからであると考えられる。また，対象とする構造物が細長い場合には風方向よりもむしろ風直交方向の外力変動の方が大きくなる。このように，風の流れに応じて空気力が発生するのであるから，その流れを規定する構造物の形状を変えることによって外力は大きく変化する。したがって，構造物の建設地点を決定するだけでは風による外力を仮定することはできない。また，流れを規定するという面では近くに他の構造物があるか否かによっても変化するし，他の構造物がある場合にはそれがどの位置にあるのか，つまり対象とする構造物の前にあるのか，後にあるのか，あるいは横にあるのかによっても大きく変わる。

一方，マクロ的な見方をすると，構造物の建設地点の違い，つまり，海岸か都市かといった違いによって風自体のプロフィールが異なる。

このように，風によって構造物に働く力を仮定することはかなり難しく，現段階では風洞実験に頼る部分が多い。風洞実験では人工的に風を発生させ，風洞内に設置した構造物模型に作用する「力」を計測する。この場合，模型基部に荷重計を設置して模型全体に加わる変動力を直接計測したり，模型各部の変動風圧力をきめ細かく計測し，その積分値から「力」を推定したりする。このようにして推定する「力」は，先に述べたように構造物まわりの空気の流れと密接に関係している。この流れの様子を実験的にとらえたのが**図10，11**であり，風洞内に長方形の剛体模型を設置し，左側を風上として風を流した結果である。図10では長方形模型1体を置き，図11ではその前に背の低い剛体模型を置いた。両者を比較することにより，試験体の設置状況により模型まわりの空気の流れに差が現れる，つまり模型に働く「力」に差が現れることがわかる。

また，この実験結果を効率良いデータとするためには，「写真」でとらえるだけではなくコンピュータを用いたデータ処理が必要である。

一方では数値解析により空気の流れを計算し，構造物に作用する力を求める方法もある。3次元解析により求めた超高層構造物まわりの空気の流れを**図12**に，後方の構造物に働く風圧分布を色の変化で表現したものを**図13**に，膜構造物に働く風圧分布を**図14**に，構造物まわりの風圧分布を等圧線で表現したもの

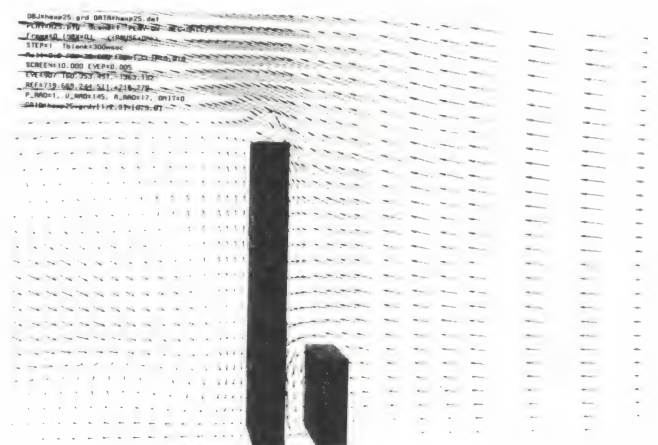



図12　解析により求めた空気の流れ

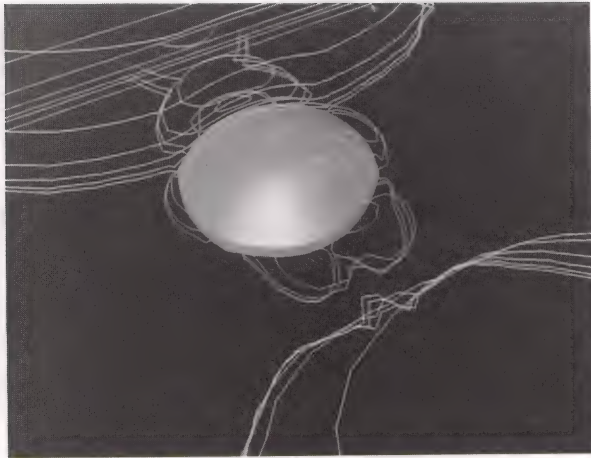



図 15　膜構造物まわりの風圧分布

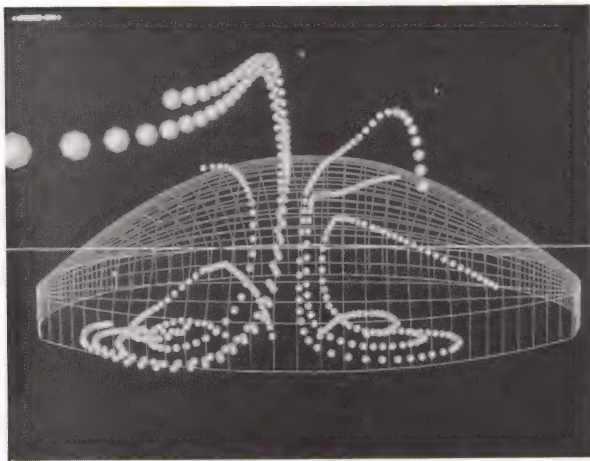



図 16　膜構造物後方の空気の流れ

を**図 15** に，構造物後方の空気の流れを**図 16** に示す。このように解析的手法を用いることにより，流れの状況ならびに構造物に働く風圧力をきめ細かく求め，さまざまな表現で見ることができる。しかし，そのためには大容量の計算機を長時間回す必要があるとともに解析に用いた仮定が正しいことを確認する必要がある。したがって，実験と同じように解析途上で計算が正しく進んでいるか否かをチェックし，必要があればそのつど諸データの変更を行う方が効率的であろう。

風による構造物の揺れ

構造物が風の力を受けて振動する場合，その揺れ方はさまざまである。現在までに知られている揺れ方を大別すると，バフェッティング，渦励振，フラッタの 3 種類となる。

風の強さは時間とともに変動するが，その変動に伴って空気力が変動し構造物を揺らせるのがバフェッティングである。また，構造物まわりに風が流れると構造物の後方に渦が放出される。この渦の放出は周期的に行われるが，この周期と構造物の周期が一致すると構造物の共振が生じる。これが渦励振である。これらはいずれも風の力を外力とする強制振動である。一方，構造物がなんらかの原因で振動を始めると構造物まわりの空気の流れがその振動を助長するように働き，大きな振動を生じることがある。この振動がフラッタであり，いわゆる自励振動である。また，先にあげた強制振動のうち渦励振はフラッタ的な挙動を示すことも考えられる。これは構造物が振動を始めることにより振動源である渦自体がその影響を受け，渦の放出に変化が現れることが予想されるからである。

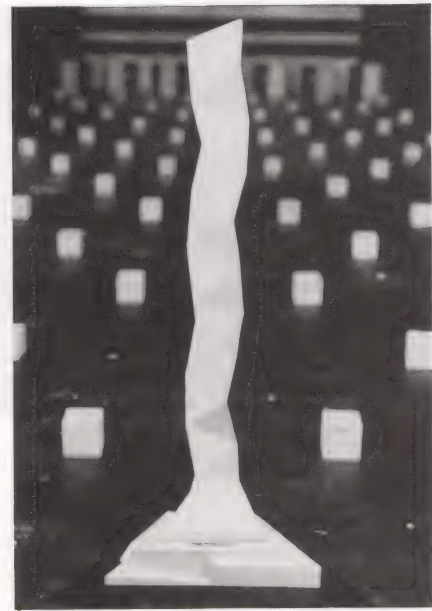



図 17　風洞を用いた振動実験例

超々高層構造物やタワーなどの設計にはこれらの振動をあらかじめ予測する必要があるが，現段階では実験に頼る部分が多い。特に自励振動を解析的に精度良く解くことはかなり難しい。一方，自励振動が発生しない領域における強制振動については，実験あるいは数値解析により外力を求め，この外力を構造物に与えることにより振動予測をすることができる。

図 17 に風洞を用いた構造物の振動実験例を示す。この構造物は稜線が三重螺旋形状をした展望塔であり，最高部高さは約 100 m である。なお，この三重螺旋形状は一稜 9.6 m の正四面体 27 個を連結することによって構成されている。このように本構造物は特殊な形状をしており，解析だけでは性状を把握しづらい面もある。したがって，振動実験ならびに構造物に作用する「力」を求める空気力実験を実施し，構造的安全性の確認を行っている。

これまで述べてきたように，実験あるいは数値解析のいずれの方法をとるにしても，振動現象を把握するためには空気の「流れ」を明確にする必要がある。「流れ」の状況を連続的にとらえ空気力を予測するとともに，その「流れ」をどう変えれば空気力を小さくできるのか，つまり振動を少なくすることができるのか興味の尽きないところである。

また「流れ」の解明にあたっては，実験および数値解析を適切に使い分けるとともに両者を対応させながら総合的に研究を進める必要がある。その場合，データ処理，画像処理を含め，コンピュータをいかに効率良く用いるかが目標を達成するための大きなキーポイントとなることが予想される。

特集3

機械設計のための計算力学

計算力学の展望—現状と将来

誕生間もない揺籃期にある計算力学について，その歴史と現状を概説し，将来の展望について述べる。

武田　洋*

はじめに

計算力学（computational mechanics）とは，応用力学に関する比較的新しい学問分野である。従来より応用力学に関する学問分野は，連続体力学，固体力学，構造力学，材料力学，流体力学，熱力学，電磁気学などのように，取り扱う力学現象に対応してその学問分野が種分けされているが，計算力学の場合はいささかニュアンスを異にする。計算力学という学問分野は計算や計算機に関する力学を取り扱うのではなく，今日まで応用力学の分野で強力ではあるが単なる道具として広く用いられてきたコンピュータを，力学現象に対する定式化から解析，さらに結果の評価・応用にいたるあらゆる段階において念頭に入れることにより，理論から応用までを統一的かつ効率的に展開しようとするものである。

ここでは，誕生間もない揺籃期にある計算力学についてその歴史，現状を要約し，将来の展望について述べる。

計算力学にいたる道のり

コンピュータが誕生する以前から，応用力学に携わってきた研究者や技術者は，種々多様な具体的な力学的問題を解決してきたことは周知のことであり，その問題解決の過程において数々の新しい数学的な解析手法が生まれていることもよく知られている。数値計算に多大な労力が必要とされていた時代では，取り扱う力学現象に対して適切な仮定を導入することにより，解析的手法を用いて問題を解決せざるを得なかったが，解析的に取り扱うことができない複雑な現象や形状をもつ問題に対しては，数値的手法を用いた解析も試みられてきた。

現在も代表的な数値解析法の一つとして，特に計算流体力学の分野で広く用いられている差分法(finite difference method)は，偏微分方程式における偏導関数を差分近似で置き換えることにより離散系の連立代数方程式を導くものであり，その起源はNewtonやEuler，Gaussまでさかのぼることはよく知られている。今日までに差分法に関する数多くの重要な研究が発表されているが，特にCourant-Friedrichs-Lewyの差分スキー

*たけだ　ひろし　　法政大学　計算センター　〒184　東京都小金井市梶野町3-7-2

ムの安定性に関する研究は，差分法の基本的考察や実用上の注意を促すうえで現在も頻繁に引用されており，差分法に関する先駆的研究として最も重要なものであろう。また，実際の問題に対する差分法の応用に関しては，川口の円柱まわりの2次元Navier-Stokes流れの解析を忘れることができない。この研究はコンピュータの存在しない環境のもとで，手回し計算機を用いて2年あまりを費やし計算を実行したものである。

差分法と同様に現在広く用いられている数値解析法の一つとして有限要素法（finite element method）がある。この方法は差分法が偏微分方程式そのもの（強形式）を近似するのに対して，偏微分方程式の弱形式（変分形式）を考え，さらにその定義域を比較的簡単な形状をした要素（有限要素）の集まりとみなし，未知関数の分布を個々の要素上で簡単な関数で仮定することにより，差分法と同様に離散系の連立代数方程式を導くものである。この方法の直接的な出発となった研究は，1950年代における航空機の構造解析に関する研究であり，これらの研究は当時実用段階に入ったコンピュータを用いて，大規模かつ複雑な形状をもつ航空機の構造解析を行うためのものであった。しかし，この方法の理論的な基礎となるものは，幽閉中のGalerkinの研究であり，またコンピュータの誕生する以前，1943年にCourantによって弾性学におけるねじり問題の解析において有限要素法の考え方が用いられている。一方，このような解析手法は，その解析過程が行列を用いて表現されていることから，マトリクス構造解析法ともよばれているが，Southwellや鷹部屋による力学問題に対する機械的な計算法は，自動的解析法という観点からは有限要素法の源とも考えることができる。

差分法や有限要素法では力学現象を支配する未知関数は局所的な分布が仮定されているが，古典的なGalerkin法（またはその一般化としての重み付き残差法）では定義域全体を台とする大域的関数を仮定する方法が普通であり，代数方程式の計算を回避するためにFourier級数に代表される直交関数系が広く用いられてきた。流体力学の分野では，スペクトル法（spectral method）という名称で，古典的なGalerkin法を一般化し，Navier-Stokes流れをはじめとする各種の流体問題に対するコンピュータ解析法が誕生している。

力学現象の表現として偏微分方程式が広く用いられているが，一方，特異性を有する問題などのように，解析的な解を求める方法として積分方程式による表現が適している場合もある。このような力学問題に対する積分方程式法を，一般的なコンピュータ解析法として再構築した方法が境界要素法（boundary element method）とよばれる方法である。

計算力学の現状

現在，コンピュータが工学における幅広い分野で日常の道具として用いられていることは周知の事実であり，その総合的環境に対してCAE（Computer Aided Engineering）という名称がつけられている。これは製品開発から設計・製図，製作にいたる一連の技術者の活動に対して理想的なコンピュータ利用環境を提供しようとするものであり，技術者の生産性向上のために工学にかかわる企業において注目されている。この考え方の中核をなすものは計算力学であり，一方で，現状で理想的なCAEシステムの構築を時期尚早としているのもまた計算力学である。ここでは，力学的理論，数値解析法，コンピュータ・インプリメンテーションおよびコンピュータ・ハードウエアの観点から計算力学の現状を要約する。

あらゆる力学的現象は非線形であるが，その挙動に対して線形性の仮定を導入できる場合もある。一般に日常製作される構造物や機器においては，この線形性の仮定の成り立つ範囲で設計されるものが多い。このような場合，計算力学にかかわる問題は一般に線形代数方程式に帰着する。連立1次方程式の解法は直接法と反復法に大別されることはよく知られているが，ウェイブフロント法やスカイライン法とよばれる大次元の疎行列に対する効率の良い直接解法の開発や，計算回数を縮小するための方程式の番号付け替えのための各種アルゴリズムの開発により，現在では直接法が広く用いられている。しかし数十万元以上の超大型の方程式に対して，過去にコンピュータの能力から数千元の方程式を解くために用いられてきた反復法が再評価されつつあり，反復に先立って行列の条件を改良するための前処理（pre-conditioned）アルゴリズムの開発が最近の話題である。

一方，実際の工学の分野で取り扱う問題は複雑な幾何形状をもっている場合が多く，その形状の認識，解析モデルの自動作成およびコンピュータ・グラフィックスを利用した解析結果の表現に対する研究やインプリメンテーションも積極的に行われている。

しかし，計算力学の観点から現在最も重要なことは，力学的な意味での解析結果の精度の保証であろう。一般に計算力学の分野で用いている手法は近似解法であり，当然のことながら解析結果の精度は解析モデルに依存する。一般に採用されている数値解析法では解の収束，すなわち解析モデルをより一層細かくしていけば真の解に近づいていくことを保証しているだけである。したがって解析モデルを作成するためには，その問題に対する技術者の経験や力学的センスが重要となり，今日のように多くの技術者が計算力学に基づくコンピュータ解析を行っている現状では，線形問題であっても精度が十分でない解析結果が日常の設計で用いられている可能性もおおいにあり得る。このような弊害を取り除き，健全な解析手法を技術者に提供するための方法として，順応型数値解析法（adaptive numerical method）の研究が進められている。これは解析モデルに含まれる近似誤差を予測し，その予測に基づいて解析モデルを再構築し，十分な精度が得られるまで自動的に解析を繰り返す方法であり，解析モデルに対する近似解だけを用いて真の解との差をいかに的確に予測するかがキーポイントである。

一般に力学現象は時間依存であり，定常状態や静的状態と仮定できる現象については空間モデルだけが考察される。時間を含む現象の解明に対しては，線形問題では固有値問題として取り扱い，一般の非線形問題に対しては空間の離散化によって得られる時間に関する常微分方程式を取り扱うことが多い。時間積分法の数値スキームの開発に関しては従来より多くの研究が

続けられているが，計算効率を高めるためのアルゴリズムとして作用素分割法（operator splitting method）や陰解法と陽解法を組み合わせた方法が最近の話題である。

作用素分割法は，連立方程式の解法を含まない陽解法のアルゴリズムを用いて陰解法の無条件安定性を得ようとするものである。一般に陽解法ではそのアルゴリズムに連立１次方程式の求解過程を含まないが，時間刻みに対して条件付き安定となり，陰解法アルゴリズムを用いた場合と比較すると非常に多くの時間ステップを必要とする。しかし，衝撃問題のような非常に高速な力学現象に対しては，陰解法を採用したとしても空間モデルに対応した適切な時間増分を必要とし，陽解法と比較すると計算効率が悪いのが普通である。だが実際に解明したい問題では，複雑な現象が組み合わさっており，その複雑な力学現象に対応して最も効率の良い時間アルゴリズムを採用することが理想である。そのようなアルゴリズムを構築するための第一歩として，種々の陰解法スキームと陽解法スキームを任意に組み合わせた数値解法や，時間増分に対する順応型アルゴリズムおよびそのインプリメンテーションの研究が進められている。

現在，計算力学にかかわる研究者が最も多く携わっている問題は非線形現象の解明であろう。非線形問題に対する数値計算法としては古くから Newton 法が用いられており，現在でもこの単純明快な反復法が最も広く用いられている。この方法は局所的収束が保証されているだけで，構造力学における座屈現象のように不安定領域を含む場合には問題が生じる。このような問題に対しても Newton 反復法を安全に適用するための工夫として弧長増分法（arc length method）が提案されている。この方法は個々の反復過程において既知ベクトルの量をコントロールすることを基本としている。また，Newton 法では個々の反復において連立１次方程式の求解過程を含むが，これを行列の乗算で置き換える BFGS 法のような準 Newton 法（quasi Newton method）に関する研究も進められている。この方法を極言すれば，Newton 法が接線を基本とするのに対して，割線を用いて反復を繰り返すものである。この方法は非線形性の強くない問題に対して効率良く働く場合が多いが，汎用性をもつアルゴリズムの構築のためにはさらに研究が必要である。

現在のハードウエアの環境は非常に多様性に富んでおり，スーパー・コンピュータ，ミニ・スーパーコンピュータ，パーソナル・スーパーコンピュータ，EWS，パーソナルコンピュータなどさまざまな機種が開発されているが，スーパー・コンピュータの能力は，計算力学における大規模な問題に対しては十分であるとはいえないのが現状である。しかし，数年前と比較すると技術者に与えられているコンピュータ環境は格段に改良されていることも事実である。現在，総合的なコンピュータ環境の中で計算力学にかかわるコンピュータ利用をどのようにしたらよいかという研究が進められているが，非常にダイナミックな状態のもとで唯一の理想的利用形態を提案することは不可能に近い。現在計算力学の分野でハードウエア・アーキテクチャに最も関連した話題は，ベクトル処理や並列処理の機能をもつスーパー・コンピュータの能力を十分に生かすようなアルゴリズムを構築するために，力学的問題の定式化から計算にいたる過程を見直す試みである。

計算力学におけるコンピュータ・グラフィックスの役割

計算力学応用のためのコンピュータ・プログラムの構成は，力学解析自体にかかわる部分と解析の前後処理にかかわる部分に大別される。解析のための前処理プログラムは，解析モデルの自動生成や作成されたモデルのチェックを行うためのものであり，プリ・プロセッサとよばれる。解析結果に対する後処理プログラムは，結果を図表示したり設計基準などにのっとり解析結果を評価するためのもので，ポスト・プロセッサとよばれる。実際の総合的な環境の中に解析プログラムを健全に位置付けるためには，他のシステムとの密接な連結や解析にかかわる期間の最小化が必要不可欠となる。このための中心的な役割を果たすものがプリ／ポスト・プロセッサであり，計算力学の実用化が高まるにつれて多くの関心を集めている。

プリ／ポスト・プロセッサの主要部分は，幾何学的形状の認識や力学現象の解釈・評価に関するものであり，このために生の数値データを直接用いることは非常に効率が悪く，なんらかの加工が必要となる。形状や現象を全体的に把握する際に古くから広く用いられている方法は，人間の視覚に訴える可視化であることはいうまでもない。コンピュータを利用した可視化の分野はコンピュータ・グラフィックスという名前で総称され，現在は工学に限らずあらゆる分野で脚光を浴びており，これにはコンピュータ本体の急速な進歩・発展に加えて，パーソナルコンピュータに代表されるような廉価なグラフィックス装置の急激かつ広範囲な普及が大きな役割を果たしていることは周知の通りである。

計算力学の分野におけるコンピュータ・グラフィックスの応用は，解析する物体やその計算力学モデルを認識するための幾何学的形状の表現と，計算力学解析によって得られた力学的現象の可視化に大別される。

ここでは，計算力学におけるコンピュータ・グラフィックス利用の個々についての解説に先立って，解析から結果の評価にいたる一連の過程を順に考えてみる。第１は，解析に先立つ解析モデルの形状の認識とその検証であり，一般に標準的なソリッド・モデル表示が用いられており，このような表現は特に複雑な形状をした問題に対しては不可欠となる。力学現象の解析においては離散モデルが用いられるが，この検証や解析結果として得られる形状変化を調べるために変形前後の離散モデルの図化が用いられる。解析結果を評価するためには，スカラー変数に対しては等高線表示やカラー表示が一般的に用いられている。ベクトル量のように方向性をもつ変数に対しては，ベクトル線表示によりその方向と大きさの分布を評価する。さらに代表的変数の時刻歴，応力—歪み関係のような２変数の関係，着目する線に沿った変数の変化などの評価に関してはグラフ表現が用いられる。解析結果は純粋な工学的評価のためだけでなく，製品パンフレットや顧客への説明資料として用いられる場合もあり，このようなプレゼンテーション用としてはソリッド・モデル図と解析結果を重ね合わせたような表現などが用い

られている。

■形状の可視化

一般的な図形をコンピュータを用いて表現するためには2つの方法がある。第1はソリッド・モデリングとよばれる方法で，形状を立方体，円柱，球，円錐などのプリミティブとよばれる基本構成要素の論理演算によって構成する方法であり，第2は多角形メッシュから図形を構成する幾何モデリングとよばれる方法で，線画により図形を表現することができるのでソリッド・モデルと対比してワイヤー・フレーム・モデルともよばれる。

形状の表現において多くの計算時間を必要とする困難な問題は，人間が物体を観察した場合と同様に目に見えない部分を表さないようにするための隠線（面）消去法である。隠線消去法は最終的に図形をCRTなどの図形出力装置に写像する方法であるが，ソリッド・モデリングの方法では観察者と表現する物体の間に1つの面を想定し，その面（CRT上）の1つの点（ピクセル）ごとに対応する物体の位置を定め，それを表現する光線追跡法とよばれる方法が一般に用いられている。

隠線消去のために提案されているさまざまなアルゴリズムは標準的な教科書に詳しく解説されているが，解析モデルを効率良く表現するための研究も進められている。例えば，角度や三角関数の演算を回避し，効率の良いデータ構成を用いた簡単かつ効率の良いアルゴリズムの研究や，1つの線をその近傍の面についてだけ比較することによって計算時間がモデルの総面数に比例するようなアルゴリズムの提案などがある。

形状やモデルの認識のために最近用いられている表現方法としては，光の透過性を表現する方法などが計算力学の分野で応用されている。物体のコンポーネントごとに光の透過特性を変え，実際には目に見えない細部までも表現することによって形状や解析モデルを検証したり，変形前後の形状を重ね合わせて表現することによって，迅速かつ明確に解析結果を評価する方法などがある。

■力学現象の可視化

計算力学はさまざまな力学現象を解明するために用いるものであるから，当然のことながら解析によってその力学的な挙動が定量的に定められる。解析によって定められる代表的な力学的諸量は，温度や圧力のように一つの数値として表されるスカラー量，速度や力のように大きさと方向によって表されるベクトル量，および応力や歪みのように想定する方向によってその大きさと方向が異なる二階のテンソル量に大別される。一般に広く用いられている方法はスカラー量に対する可視化方法であり，ベクトル量やテンソル量に対しても，その特性をスカラー表現するための不変量（ベクトルの大きさ，相当応力，相当塑性歪みなど）を用いてスカラー量に対する可視化の方法を利用する場合が多い。スカラー量の分布を表現するために古くから用いられている方法は等高線表示である。解析モデルの等高線表示については，個々の要素を三角形に分割し，その三角形の中で変数を線形変化するものとして等高線を作成するアルゴリズムが最も単純であることから広く用いられているが，解析モデルに対する等高線を自然な形に表現するために曲線表示を導入するアルゴリズムなどが提案されている。グラフィック・ハードウエアの進歩・普及により，スカラー量の分布表現にカラー表示が最近急速に用いられている。

2次元領域のスカラー量を表示するための別の方法としては，スカラー量$f(x, y)$をz座標と考え，それを3次元的な鳥瞰図によって表現する方法がある。このような表現は通常の隠線消去のアルゴリズムを用いて表現できるが，特に有限要素モデルに対して非常に効率良くこの処理を行うためのアルゴリズムも提案されている。解析によって得られたスカラー，ベクトルおよび二階のテンソル場の全体的な挙動を容易に把握するために，その方向性と大きさを同時に可視化する方法も提案されている。

将来の方向

現在計算力学とよばれている分野の研究者は，主に流体力学にかかわる分野と構造力学や材料力学に代表される固体力学にかかわる分野に大別され，それぞれの領域で積極的な情報交換が行われているが，互いに多くの共通点をもつと同時に相互に影響を与え合っている。例えば，差分法における上流スキーム（upwind scheme）は有限要素法ではPetrov-Galerkin定式として一般化されているし，差分メッシュの代数的生成法（algebraic generation method）では有限要素法の考え方が用いられている。さらに，それぞれの分野でさまざまな現象が連成した問題に対するチャレンジが試みられている。原理的にはあらゆる力学現象は連成しているが，今日までの解析ツールではそれらを連成させて解くことは非常に限られており，一般には他の力学的現象に対しては境界条件として取り扱われている。しかし，技術革新やコンピュータ能力の向上によって，さまざまな複雑な現象の解析が必要となろう。

一方，現在のコンピュータ解析では総合的な意味での解の信頼性は保証されていない。前節で順応型数値解析法について要約したが，計算力学が広範な領域に対する健全な解析ツールとして用いられるためには，総合的な意味で順応型であるべきである。すなわち，定義された力学現象に対して最も適した力学モデルを選択し，それに対して最も効率の良い解析手法により，複合したハードウエア環境の中で最適なハードウエアを用い，その結果を適切に表現するという一連の自動化を目指す必要がある。このような総合的システムの構築に対してAIを応用しようとする試みもあるが，それ以前に純粋に計算力学の範囲で解決しなければならない問題がたくさん残っていると思われる。

おわりに

ここでは，計算力学の現状および将来の方向について概説した。ここでの解説は計算力学に関するあらゆる側面を論じたものではなく，その一部について筆者の思いつくままに記したものであることをお断りしておく。なお，この解説が本誌で企画されている計算力学に関する一連の具体的な最先端技術の解説に対する導入となることを望んでやまない。

スーパー・コンピュータによる大規模構造解析システム

スーパー・コンピュータのハードウエアの特徴を生かした計算技法やソフトウエア，およびプリ／ポスト・プロセッサを含めた大規模構造解析システムを紹介する。

三好 俊郎*

●図１は次ページ。

はじめに

スーパー・コンピュータを用いて科学技術計算を行うことをスーパーコンピューティングという。効率的なスーパーコンピューティングの実現のため，スーパー・コンピュータの利用技術に関する研究が精力的に行われている。有限要素法(FEM)，境界要素法（BEM)，差分法（FDM）などの計算力学の分野においても，スーパー・コンピュータを用いた大規模かつ高速な解析が関心を集めている。

そこで本稿では，スーパーコンピューティングを効率良く行うための重要なポイントである高速化技法，システム・インテグレーションについて述べる。高速化技法としては，特に大規模な連立一次方程式の高速解法に焦点をあてて述べる。また，有限要素法による３次元構造解析プログラムの各種スーパー・コンピュータによるベンチマーク・テスト[1]の結果などを紹介する。

高速化技法

(1) 剛性方程式の解法

スーパー・コンピュータによる有限要素解析の利点は，高速性と大規模性の２点にある。筆者らの経験では，有限要素法による構造解析においては計算時間の約70％程度が剛性方程式(連立一次方程式)の解法に費やされる。したがって，剛性方程式の解法は有限要素解析のキーポイントである。そこで，スーパー・コンピュータによる高速計算および大規模計算に適した剛性方程式の解法（ソルバー）であるスカイライン法，ICCG法について述べる。

(2) パラレルスカイライン法

スカイライン法は直接法の一つであり，剛性マトリクスをコレスキー分解（LDL^T分解）して解く方法である。スカイライン法の計算主要部はスーパー・コンピュータが得意とする内積計算であり，高い加速率が得られる。

スカイライン法を改良し，複数のベクトル演算器を並列に動作させ，より高速性を得るために開発されたソルバーがパラレルスカイライン法[2]である。パラレルスカイライン法は，コレスキー分解において３行３列同時分解（３次元構造解析の場合）を行うものであり，高速計算に適した解法である。

(3) ICCG法

ICCG法[3]は反復法の一つであり，共役勾配法に前処理として不完全コレスキー分解を施し，収束までに要する反復回数を大幅に削減することにより高速化を達成した解法である。ICCG法は，剛性マトリクスの非零成分のみを用いて計算を行うことができるためメモリ効率が良く，大規模計算に適している。

図１は，東京大学大型計算機センターにおいて，主記憶28.5Mバイト，外部記憶128Mバイトが使用できるという環境を想定して作成したソルバーの適用範囲である。ICCG法を用いることにより約20,000節点までの大規模解析が可能となる。

(4) 内積形式のスカイライン法

今後のスーパー・コンピュータの発展の方向は，複数のCPUを有する並列方式のスーパー・コンピュータである。そこで，並列コンピュータ向きのソルバーとして内積形式のスカイライン法[4]を紹介する。

従来のスカイライン法は，縁どり法[5]とよばれるアルゴリズムを用いている。縁どり法の場合，並列性は明示的ではない。一方，内積形式のスカイライン法では，最内側DOループはベクトル処理が可能であり，さらに一つ外側のDOループに関し

*みよし としろう　東京大学 工学部精密機械工学科 〒113 東京都文京区本郷7-3-1

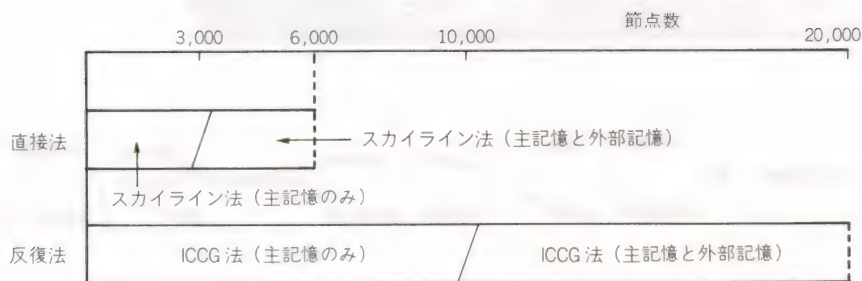
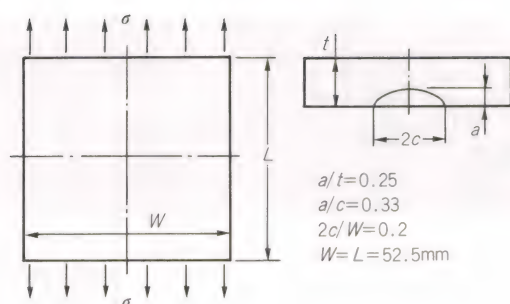



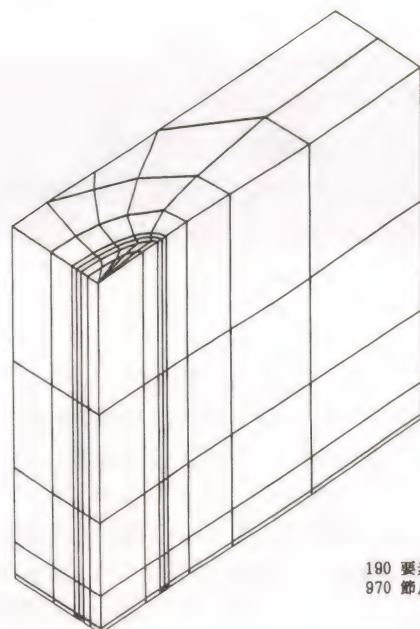
図1　3次元構造解析におけるソルバーの適用範囲

図2　3次元表面亀裂

図3　要素分割図

て並列化もできる。したがって，ベクトル化と並列化の相乗効果による高速化に適している。また，内積形式のスカイライン法においても前述のパラレルスカイライン法と同様に3行3列同時分解による高速化が有効である[4]。

(5)　ベンチマーク・テスト

スーパー・コンピュータ用に作成した有限要素解析プログラム FEM3D/S[2]および FEM 3D/I[3]を用いて，各種スーパー・コンピュータおよび汎用コンピュータによるベンチマーク・テストを行った結果について述べる[1]。FEM 3D/S および FEM 3D/I は，おのおのソルバーとして前述のパラレルスカイライン法および ICCG 法を使用している。このベンチマーク・テストは，日本機械学会 RC-74「非線形有限要素法の応用研究分科会」第4小委員会スーパーコンピュータワーキンググループの活動として行われた。

ベンチマーク問題は，**図2**に示す3次元表面き裂材の一様引張りとした。**図3**に要素分割図を示す。要素数 190，節点数 970 である。FEM 3D/S による計算時間を**表1**に，FEM 3D/I による計算時間を**表2**に示す。表1,2は左半分がスーパー・コンピュータ，右半分が汎用コンピュータによる計算時間である。表1の CRAY X-MP/3 は CRAY X-MP/4（4 CPU 構成の CRAY X-MP）の3 CPU を用いて計算したことを示している。また，ベクトル化率の（　）は推定値である。表2の MicroVAX II は EWS クラスのコンピュータである。表1からは以下のことがわかる。

① FEM 3D/S を用いた場合，スーパー・コンピュータは汎用コンピュータと比較して2〜100倍の計算速度を有する。

② FEM3D/S のベクトル化率は 92〜97%である。ベクトル化率とは，プログラムの全演算に対するベクトル処理される演算の比率のことである。一般に，ベクトル化率が 90%を超えて初めて十分な高速性が得られるといわれている。

③ FEM3D/S のスーパー・コンピュータにおけるスカラー演算時間とベクトル演算時間の比（加速率）は 3〜15 である。

また，表2からは以下のことがわかる。

① FEM3D/I を用いた場合，スーパー・コンピュータは汎用コンピュータと比較して2〜45倍の計算速度を有する。

② FEM3D/I のベクトル化率は 93〜94%である。

③ FEM3D/I のスーパー・コンピュータにおける加速率は4〜5である。

④ ICCG 法はメモリ効率が良いため Micro VAX II（主記憶 5 M バイト）でも実行可能である。

表1，2を比較すると以下のことがわかる。

表1 FEM3D/S(パラレルスカイライン法)による計算時間

機　種	コンパイラ	計算時間(秒)	ベクトル化率(%)	機　種	コンパイラ	計算時間(秒)
CRAY X-MP/1	CFT77 Rel.1.3	8.52	(92)	FACOM M-380Q	FORTRAN77	174.45
CRAY X-MP/3	CFT77 Rel.1.3	3.34	(92)	FACOM M-380Q	VS FORTRAN	203.12
FACOM VP-50	FORTRAN77/VP	15.02	96.09	IBM 3081K	FORTRAN77	311.41
FACOM VP-100	VP COMP.V10/L20	10.11	93.50	IBM 3081K	VS FORTRAN	337.58
HITAC S-810/20	FORTRAN77/HAP	7.30	96.60	IBM 3090-200	VS FORTRAN	101.68
				HITAC M-682H	OFORT77	61.59
				HITAC M-682H(IAP)	OFORT77	25.06

表2 FEM3D/I(ICCG法)による計算時間

機　種	コンパイラ	計算時間(秒)	ベクトル化率(%)	機　種	コンパイラ	計算時間(秒)
CRAY X-MP/1	CFT77 Rel.1.3	14.09		DEC Micro VAX II	FORTRAN V4.6	8603.19
FACOM VP-50	FORTRAN77/VP	24.26	93.22	FACOM M-380Q	FORTRAN77	224.67
FACOM VP-100	VP COMP. V10/L30	21.40	94.40	FACOM M-380Q	VS FORTRAN	226.30
HITAC S-810/20	FORTRAN77/HAP	16.27		IBM 3081K	FORTRAN77	310.16
HITAC S-820/80	FORTRAN77/HAP	6.75		IBM 3081K	VS FORTRAN	309.06
				IBM 3090-200	VS FORTRAN	99.83
				HITAC M-682H	OFORT77	57.98
				HITAC M-682H(IAP)	OFORT77	45.84

1） ベンチマーク問題のき裂材の引張りは，ICCG法が不得意とする収束性の悪い問題であるにもかかわらず，スカラー演算の演算時間はFEM3D/S, FEM3D/Iともにほぼ等しい。筆者らの経験によれば，性質の良い問題（例えば，立方体の一様引張り）に対してはICCG法の方がスカラー演算時間は短い[6]。

2） FEM3D/SとFEM3D/Iでは，ベクトル演算時間はFEM3D/Sが短い。これはスカイライン法がICCG法よりも加速率が大きいためである。ICCG法は前処理によって演算量を減らし高速化しているのに対し，パラレルスカイライン法は加速率が大きいため高速化している。

3） メモリ効率はICCG法がスカイライン法より優れているため，超大規模問題の計算にはICCG法が有効である。

以上の結果より，FEM3D/S, FEM3D/Iで用いた高速化技法は一般性があり，複数のスーパー・コンピュータにおいて有効であることがわかる。また，FEM3D/S, FEM3D/Iとスーパー・コンピュータを用いた場合，汎用コンピュータと比較して最高100倍程度の計算速度が得られる。さらに，精度についても検討を行い，スーパー・コンピュータのベクトル演算による精度低下がないことを確認している。

システム・インテグレーション

スーパー・コンピュータを用いることにより解析できる問題の規模は大きくなり，より精密な解析ができるようになる。その結果，従来より入出力データ量は増加する。入出力データの扱いに要する時間が，スーパー・コンピュータによる計算時間に比べてはるかに長くては，スーパー・コンピュータの価値は半減する。スーパーコンピューティングは単にスーパー・コンピュータによる高速数値計算のみでは完結しない。効率的なデータ入出力システム（プリ／ポスト・プロセッサ）の開発が必要である。

スーパー・コンピュータおよびこれに接続される端末などを含むコンピュータシステムの有効利用に関する事柄は，システム・インテグレーションとよばれている。効率的なスーパーコンピューティングを実現するために，システム・インテグレーションに関する研究が不可欠である。

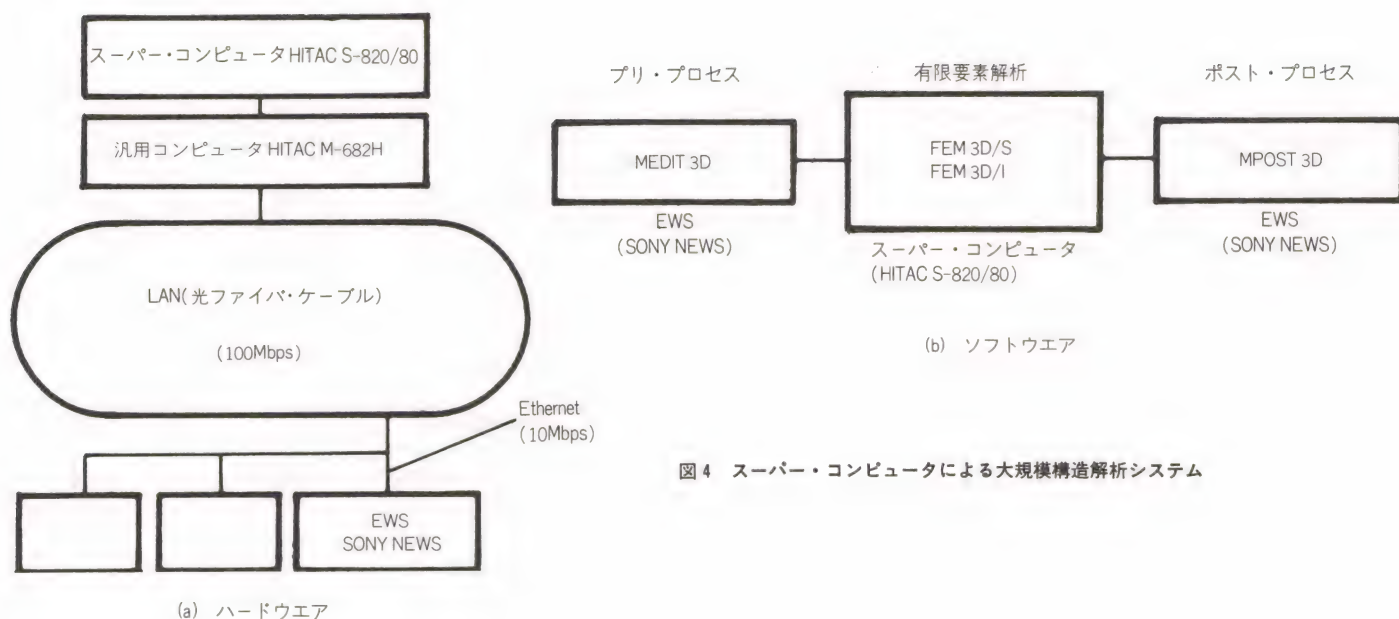



図4　スーパー・コンピュータによる大規模構造解析システム

図4に，スーパー・コンピュータによる大規模構造解析システム[7]の例を示す。本システムでは，有限要素解析にスーパー・コンピュータ HITAC S-820/80 を，プリ／ポスト・プロセッサに EWS を用いている。EWS は SONY NEWS NWS-830 である。EWS とスーパー・コンピュータは LAN により接続されている。

本システムを用いて，**図5**に示す軸方向圧縮を受けるクランク材の弾塑性解析を行った。要素数 3,456，節点数 17,605 の大規模な解析である。解析は4ステージ行い，計算時間はスーパー・コンピュータ HITAC S-810/20 を用いて 1764 秒である。解析結果は，ポスト・プロセッサを用いてカラー・グラフィックス表示する。解析結果をビジュアルにとらえることにより，評価を効率良く行うことが可能となる。

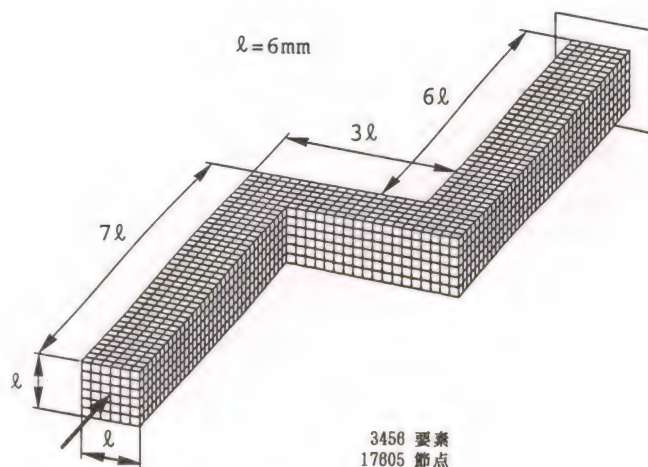



図5　クランク

おわりに

スーパー・コンピュータによる科学技術計算は日に日に増加している。効率的なスーパーコンピューティングを実現するためには高性能のハードウエアだけではなく，ハードウエアの特長を考慮した高速計算技法とソフトウエア，およびスーパー・コンピュータの計算能力を有効に利用するためのシステム・インテグレーションなどの利用技術が必要である。

本稿ではふれなかったが，身近な並列コンピュータの例として，パーソナルコンピュータに付加するかたちで用いるトランスピュータなどもある。トランスピュータを4台用いた疎結合並列コンピュータにおいて，前述の内積形式のスカイライン法を用いることにより，EWS の 10～20 倍もの高速な有限要素解析を行うことができる[4]。一方，演算速度に加えて高度なグラフィックス能力を有するグラフィックス・スーパーコンピュータの登場も興味深い。これを用いて解析結果をアニメーション化することにより，さらに効率的なシステムの構築が可能になると思われる。

参考文献

1） 三好俊郎，吉田有一郎，高野直樹：「スーパーコンピュータ用三次元構造解析プログラムのベンチマークテスト」，日本機械学会第 955 回講演会・計算力学講演概要集，pp.13-14.1989.1
2） 三好俊郎，吉田有一郎：「スーパーコンピュータによる三次元表面き裂の有限要素解析（第2報）」，日本機械学会論文集（A編），Vol.53，No.486，pp.255-260，1987
3） 三好俊郎，高野直樹，吉田有一郎：「スーパーコンピュータによる大規模構造解析」，日本機械学会論文集（A編），Vol.53，No.492，pp.1607-1613，1987
4） 三好俊郎，高野直樹：「並列コンピュータによる有限要素解析の高速化」，日本機械学会第 955 回講演会・計算力学講演概要集，pp.18-19，1989.1
5） 村田健郎，小国力，唐木幸比古：「スーパーコンピュータ」，丸善，1985
6） 三好俊郎，高野直樹，吉田有一郎：「スーパーコンピュータによる三次元有限要素解析」，日本鋼構造協会構造工学における数値解析法シンポジウム論文集，Vol.10，pp.287-292，1986
7） 三好俊郎，吉田有一郎：「スーパーコンピュータによる大規模構造解析システムの開発」，日本機械学会第 955 回講演会・計算力学講演概要集，pp.11-12，1989.1

形状最適化設計
―強いかたちと自由曲面

強いかたちとは何か? 機械・構造物の形を工夫すると，材料を節約し，軽くて強い物をつくることができる。強さに基準をおく形状最適化と自由曲面について述べる。

多田 幸生*

はじめに

昨今，丸みを帯びたものや流線形の商品の出現が話題である[1]。曲線の暖かさ，女性の身体のような柔らかさ，優雅さとか，逆モダンのレトロブームに乗って曲線あるいは曲面を売物にしたデザインがはやっている。しかし，これは単に角ばったものに対抗してのいわゆるアート的なデザインの側面からきたものとみられているが，それらの中には今日の技術の成果の現れとなっているものもある。よく知られているように，スポーツカーのボディーや飛行機の翼のように，風や空気抵抗に抗してその最大の機能を発揮するために多くの先人がその最適なかたちを求めてきた。これらは，その目的が機能的最適化であったにもかかわらず非常に美しい形をしている。シダールは，彼の著書[2]の緒言で“最善の可能な設計をなされたものは美しいだろう”と述べているが，まさにその通りであるという気がする。

さて，最適化設計には機能的なもの，強度的なものなど，種々の目的があるが，それらが所要の目的を達しているかどうかを評価，判断するには，シミュレーションによる現象・挙動の予測が必要である。近年の電子計算機の発展は，これらの基礎となる大規模な数値計算を可能ならしめてきた。さらに，こうして最適化された形態が実際に市場に出現するには，複雑な自由曲面を安価に加工製作できるようになった加工・生産技術の進歩に負うところが大きい。

本稿では，最適性の基準を主として強さにおく形状最適化設計について述べる。古来，手に入る材料を有効に利用して丈夫で安全な建物・機械などの構造物を造るのに，経験と自然に学ぶことから，その形を工夫することがなされてきた[3),4)]。机・椅子の脚，アーチ形の構造，船のマスト，パイプ型構造，トラスとできるだけ安くて，あるいは軽くて強い構造物を求め，設計者の努力がなされてきたが，その基本は強度解析（構造解析）である。したがって，まず，強いかたちとは何か，そして計算力学でどのような形状最適化が，どのように行われているかについて紹介する。また，それと関連して自由曲面についても少しふれることにする。

強いかたち

負荷荷重や材料の強さを評価する基準として応力がある。これは**図1**のように，ある面積が A の面に垂直方向から引っ張りあるいは圧縮の荷重 P が作用するとき，その単位面積当たりの力，すなわち

$$\sigma(\text{垂直応力})=P/A \tag{1}$$

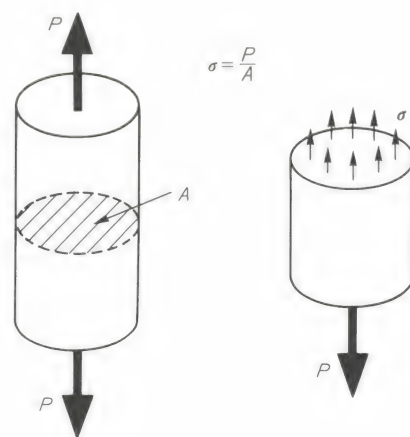



図1 引張問題と垂直応力

*ただ ゆきお 神戸大学 工学部 システム工学科 〒657 兵庫県神戸市灘区六甲台町

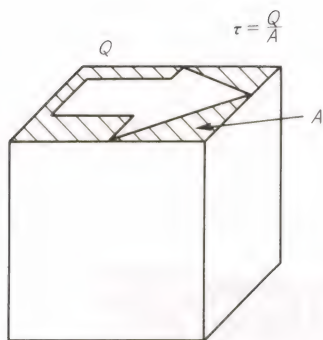



図2　せん断応力　　図3　ねじり問題

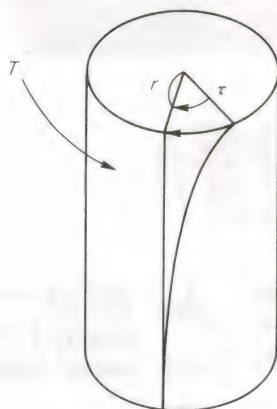



図4　片持ちばりの曲げ問題

として定義される。一方，**図2**のように，面に平行に面をずらすように作用する力 Q に対しても，同様に単位面積当たりの力として

$$\tau(\text{せん断応力}) = Q/A \tag{2}$$

が定義される。垂直応力とせん断応力は面に対する作用方向が異なっている。

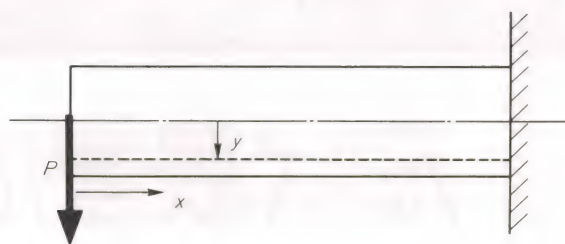
同じ荷重 P で引っ張られても，細いひもと太いひもではダメージが異なる。荷重の大きさを徐々に上げていくと，ひもが同じ材料でできているかぎり明らかに細いひもの方が早く切れるはずである。これは，上で定義した応力というものでみれば説明がつくであろう。作用する荷重による応力 σ が材料のもつ強さ(ここでは引張り強さとよんでおく) σ_0 を超えるまでひもは切れない。したがって，働く荷重の大きさがわかっているときは，σ が σ_0 より十分小さくなるように太いひもを用意すればよいことになる。このように応力が設計の目安となる。

さて，図1の場合，断面積がどこでも一定値 A であるとすると，そこで生じる応力もどこでも一定となる。したがって，切れやすさはどこでも一様のはずである。このような状態を"一様強さ"という。**図3**のように，半径 a が一定の丸い棒にトルク T を負荷すると，棒はその中心軸を固定して外側ほど回転するというねじれの変形をする。このとき棒にはせん断応力 τ

$$\tau = rT/I_p \tag{3}$$

$$I_p = \pi a^2/2 \tag{4}$$

が生じる。I_p は断面2次極モーメントとよばれるもので，棒の断面形状に関係するねじれにくさを表している（引張り力に対する断面積に対応する)。ねじりによるせん断応力は，棒の横断面内で見ると中心軸では0で，半径 r に比例して大きな値となり，棒の表面（$r=a$）で最大値となる。しかし，棒の表面上では，棒の上から下までどこでも同じ大きさである。この場合も，棒が破損する可能性は棒の長さ方向で一様であるから，やはり"一様強さ"の構造という。

次に，**図4**のような片持ちばりという構造物を考える。はりにモーメント M が作用するとき，はりは曲がり，はりの長手方向の繊維は伸縮しない中心部にある面（中立面）を境に，その上部の繊維は伸び，下部のものは縮むので，応力

$$\sigma = yM/I \tag{5}$$

が生じる。ここで，y は中立面からの距離であり，I ははりの曲げにくさを表す断面2次モーメントというものである。例えば，はりを長方形断面とし，その高さを h，幅（紙面奥行方向）を b とすると

$$I = bh^3/12 \tag{6}$$

で与えられる。また，円形断面とすると

$$I = \pi a^4/4 \tag{7}$$

で与えられる。式(5)より，はりの先端から同一距離 x の位置にある横断面内では，応力ははりの上下表面（長方形断面では $y=\pm h/2$，円形断面では $y=\pm a$）で正または負の最大値をとることがわかる。これらのことから，パイプのような中空材を用いると，断面積は小さいがそれに比して大きな断面2次モーメントをもつので，ほぼその強度を失わずに少ない材料費を実現できることがわかる。

はりの応力の長手方向の分布を見てみる。例えば図のように，これの先端（$x=0$）に荷重 P が作用するときは，モーメントは力×腕の長さ

$$M(x) = -Px \tag{8}$$

で表される。したがって，各断面の最大応力の分布は

$$\sigma(x) = Px/Z \tag{9}$$

$$Z = I/|y|_{\max} \tag{10}$$

で与えられる。ここで，Z は断面係数とよばれる。式(9)より，一様断面のはりでは固定端（$x=L$）において最大応力

$$\sigma_{\max} = PL/Z \tag{11}$$

をとることになる。長方形断面では

$$\sigma_{\max} = 6PL/(bh^2) \tag{12}$$

であるから，一様断面で設計するときは，これが許容応力 σ_a（材料のもつ強さ σ_0 を安全率（>1）で割って強度に余裕をもたせたもの）を超えないような大きさの断面形状を設定しなければならない（これを"応力制約の設計"という）ことになる。しかし，はりの先端近傍（$x=0$ 付近）ではほとんど応力が0であるのに，このような大きな断面積をもたせることは材料の非常な無駄遣いといえる。そこで，式(8)で与えられるモーメント分布に合わせて形状を変化させてやり，モーメントの大きなところでは大きい断面係数 Z を与え，モーメントの小さなところでは小さい断面係数 Z を与えることにすると，効率良く材料を使えることになる。長方形断面の場合に幅 b は一定で，高さ h を図

5のように

$$h(x)=\sqrt{6PL/(b\sigma_a)}\ \sqrt{x/L} \tag{13}$$

として変化させると，どの断面 x においても最大値は σ_a となり，“一様強さ”のはりが得られる。実際，この形は，応力制約下でのはり高さ h を変える設計の中では最も軽い構造（最も軽い構造を求めることを“最小重量設計”という）を与えている。また，式(13)に比例する形

$$h(x)=\{3V_0/(2bL)\}\ \sqrt{x/L} \tag{14}$$

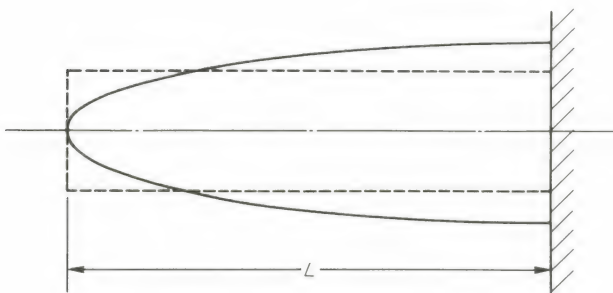
は，“一様強さ”で，かつ同一重量（体積 V_0 一定）でいろいろな形をしたはりの中で，同じ負荷（式(8)）を受けるときにそれらのはり中で生じる応力の最大値が最も小さくてすむ形でもある。このような意味で，これらの形を“強いかたち”とよぶことにする。

実際の機械などの構造物は，2次元•3次元の物体であり，前述したような垂直応力，せん断応力が2方向•3方向から作用する。したがって，強さの基準は先の単一の応力ではなく，それらを組み合わせたもので定義される。材料の性質とその使われ方で，その組合せ方・強さの基準の取り方は異なるが，それがいったん定義されると，以下の形状最適化の手法はその決め方にほぼ関係なく適用できるので，ここではこれ以上ふれないことにする。

計算力学と最適構造設計

これまで述べてきたように，“強いかたち”とは“一様強さ”をもつものとか，応力（強さ）制約内で材料を最も効率良く使う“最小重量”を与える形であるが，一般の構造物内ではその中で生じる最大応力を，前節の例のようには解析的に求めることができない。

しかし，有限要素法，境界要素法などの数値的な構造解析法の著しい発展によって，設計の対象となる構造の挙動はほとんど計算できるようになった[5)]（詳しいことは，本特集の他の記事を見ていただきたい）。

機械・構造物の設計で従来行われていた手順は，まず与えられた設計目的に応じて直感的にあるいは何かを参考にして一つの構造形状を採用し，与えられた荷重に対して変位，ひずみ，応力などの挙動を計算し，その計算された挙動がある規定された条件を満足するまで構造寸法を修正するという過程を繰り返すことであった。その条件とは，前述の応力や変位の制約の上界のこともあれば，固有振動数や座屈荷重の制限などで与えられることもある。これだけの手順により得られる構造物は普通，余裕のあり過ぎるものである。すなわち，制約は満足しているが，ここでいう“強い”構造とはいえないものである。設計においては，できるだけ材料効率の良いものを求めることが望まれるが，構造寸法の修正を試行錯誤的な単なる反復計算によって行っていたのでは，反復構造解析に非常に多くの時間を費やすだけで，なかなか最善，最適なものは得られない。

そこで，このような設計を組織的に行うという考え方が最適構造設計である。最適構造設計では，課せられた構造挙動への制限を制約条件として，体積，コストなどの目的関数によって定義される最適性の目標を最善に達成するような設計変数，すなわち最適寸法・幾何形状が探索される。その手段として，数理計画的な方法や構造物というものがもつ特徴を利用した方法が用いられてきた。文献5)の序文では，このことを「計算機の機能を最大限に利用して最適化の過程を自動化する方法と，設計者の洞察力と計算機の機能との強調をはかる方法」の2つの発展がみられると述べている。

最適構造設計のその後の理論・応用両面の発展は，この分野の国際会議の内容から抜粋された書物[6),7)]を見れば明らかである。

図5　一様強さの片持ちばり

形状最適化

このように，最適構造設計が行われるようになってきたが，まだその当初は“寸法（sizing）最適化”であった[8)]。“寸法最適化”とは，狭い意味ではトラスやラーメンなどの骨組み構造の部材寸法の最適化である。すなわち，位相関係（部材どうしの結合関係およびその結合点の位置）は固定して，鉄塔や橋などの骨組み中の各部材の最適な断面積を決める設計である。実際の骨組み構造は非常に多くの部材からなるので，信頼性設計[9)]など寸法最適化だけでも数理計画・計算力学的に非常に複雑な問題である。もう少し広い意味では，はりとか柱などの棒材の断面積分布の最適化，板材の板厚分布の最適化である。前述のはり形状の最適化がこれにあたり，これはある意味では形状の最適化といえる。

機械・構造設計においてその幾何形状を設計する場合，ある基本形状を仮定して高さ，幅など，その形状を代表するいくつかの寸法（製図における寸法に対応する場合が多い）を設計変数として採用し，それを最適化することで形状最適化が図られている。最適構造設計がCADに組み込まれるようになってきているが，まだこのような取扱いが主流と思われる。

これらの寸法変数を設計変数とするものに対して，計算機の発展とともにその実用性がでてきたのが，形状そのものを設計変数とする形状最適化である。構造物を有限要素法や境界要素法で離散化して，その節点の座標そのものを設計変数とするものや，物体の表面形状をいくつかの既知の関数の重み付き和で表し，その重みを設計変数とするものなど，形状設計変数の取り方にはいく通りかみられるが[8),16)]，いずれもあまり初期形状には依存せず，自由な曲面が得られるのが特徴である。形状

(a) 解析モデル形状

(b) 荷重条件

(c) 有限要素モデル

(d) 横断面形状

(e) 縦断面形状

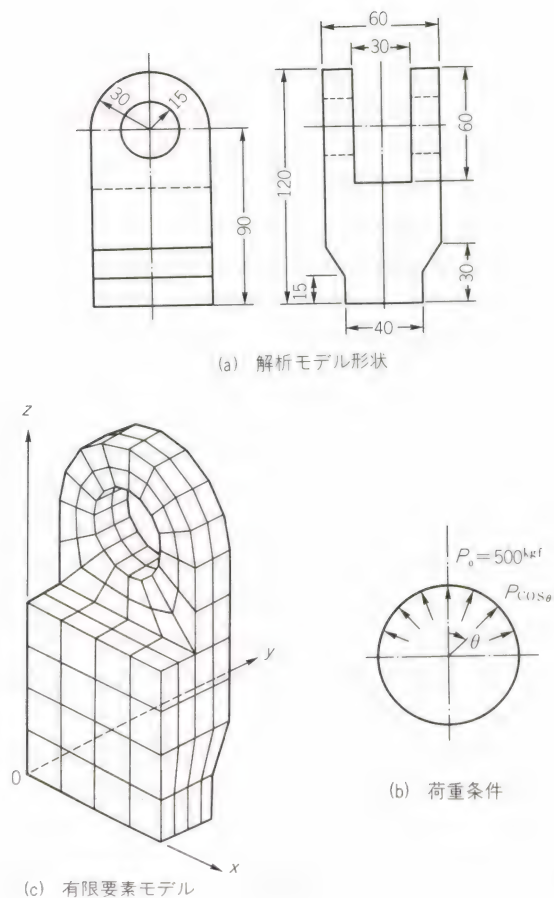



図 9 継手

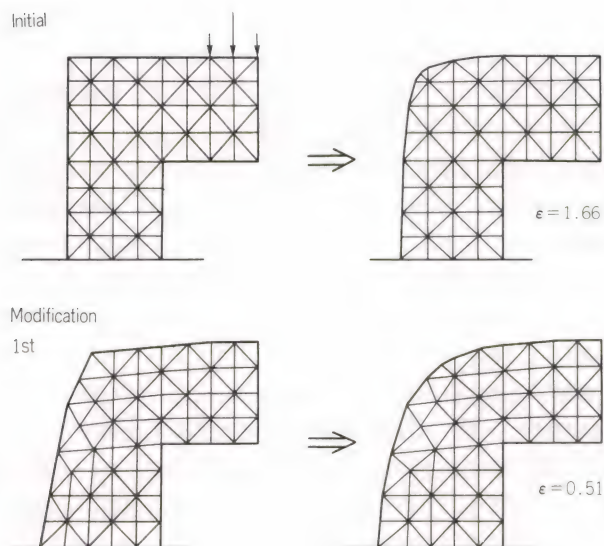
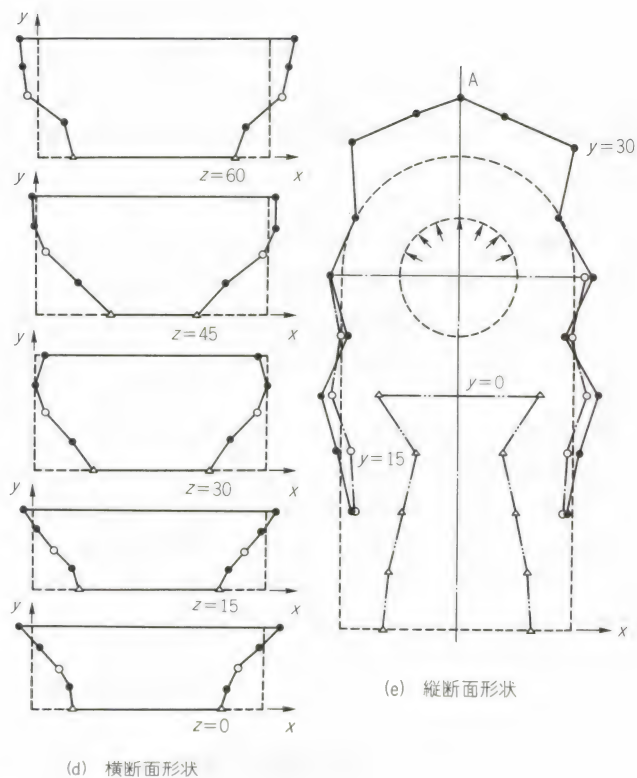



図 6 Γ 字型構造物（有限要素法）

図 7 Γ 字型構造物（境界要素法）

最適化の手法には数理計画法を利用するものが多いが，応力などの力学的諸量と局所寸法の関係を利用し，厳密な感度計算（ある設計変数の変更が応力や変位などに対してどのような影響を与えるかを計算すること）を行わずに設計変更量を求める方法がいくつか提案されている。

有限要素法では，構造解析の問題は連立方程式

$$\boldsymbol{KU}=\boldsymbol{F} \tag{15}$$

のかたちで表される。$\boldsymbol{K}$ は剛性行列とよばれ，構造物の形と材料の性質から求められる。$\boldsymbol{F}$ は節点荷重ベクトルで，与えられた外力と構造形状などから決まる。$\boldsymbol{U}$ は節点の変位を並べたベクトルであり，これを計算することで応力などの他の構造挙動

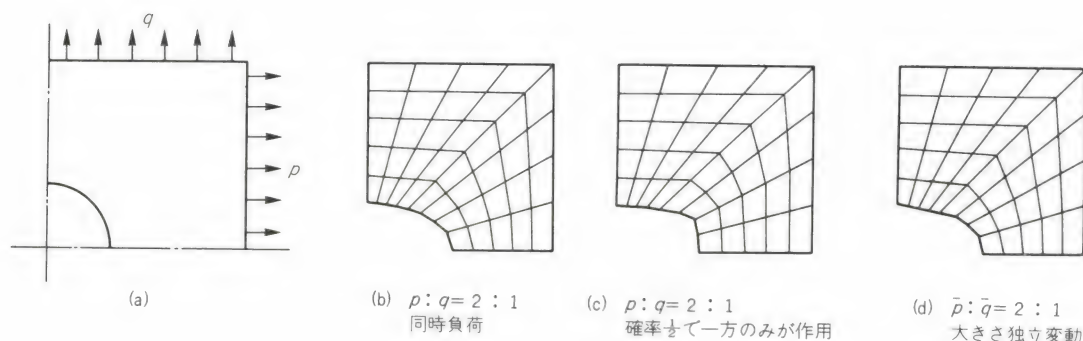



図10 有孔平版の孔の最適化

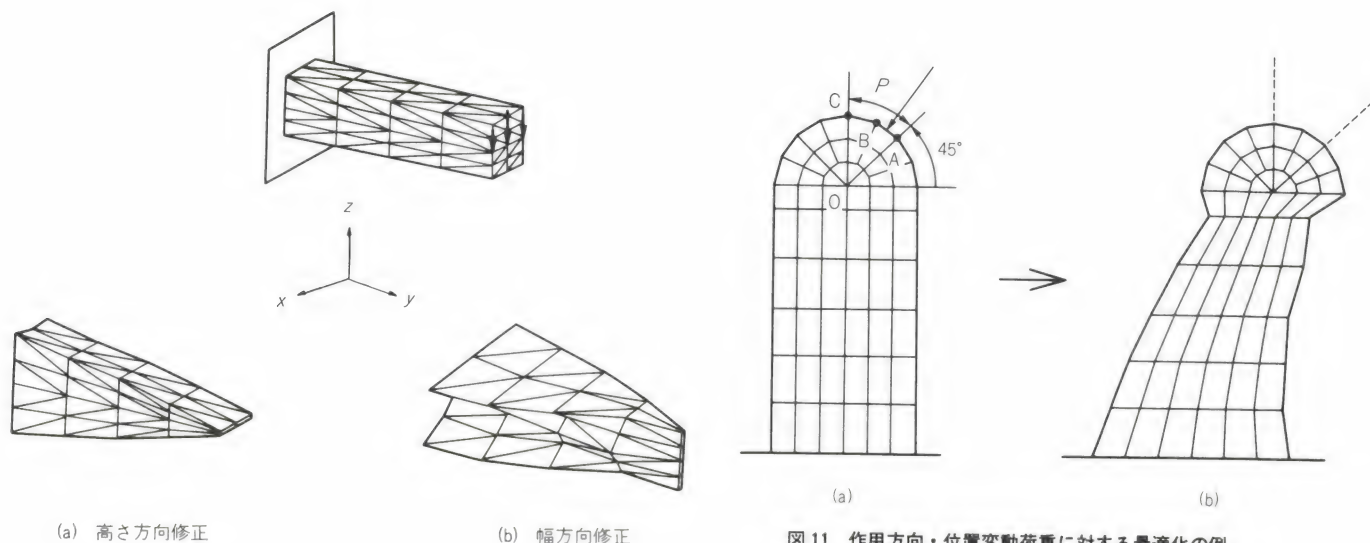



図11 作用方向・位置変動荷重に対する最適化の例

図8 片持ちばり

が求められる。式(15)を設計変数 θ で微分し，整理すると

$$\boldsymbol{K}\mathrm{d}\boldsymbol{U}/\mathrm{d}\theta=\mathrm{d}\boldsymbol{F}/\mathrm{d}\theta-(\mathrm{d}\boldsymbol{K}/\mathrm{d}\theta)\boldsymbol{U} \tag{16}$$

となるので，形状 θ を少し変更させたときの変位 $\boldsymbol{U}$ の変化は，式(16)の右辺を仮の外力とする構造解析(すなわち，$\boldsymbol{K}^{-1}$をかけて連立方程式を解く）を行うことによって計算される。応力などの感度はこの $\mathrm{d}\boldsymbol{U}/\mathrm{d}\theta$ から導出できるので，これらの感度を用いて数理計画法を適用すればよい。

感度計算のために設計変数の数だけ式(16)を解くのはやはり相当の計算時間を要するので，力学的洞察を利用した方法は有効である。生じている応力を軽減したい分だけその比に応じて要素を膨らませるというパターン変換法[10]や，目標応力と生じている応力の差の分だけ外形状を生長させる方法[11]などは，局所的応力を設計変更量決定の基準にして一様強さの形状を得ようとするものである。

これらに対して，筆者らのグループは変分原理（連続体の力学における最適化理論）によって得られる最適性の条件から求められるある量をエネルギー密度ととらえることにより，その比を利用して形状修正していく方法を提案している[12)~15)]。**図6〜11** に筆者らの方法による形状最適化の例を示す。図6と7は，同じ対象をそれぞれ有限要素法と境界要素法で最適化したものである。図6では途中で初期形状を与え直している。この例のように，形状最適化設計では形状修正過程において形状が初期形状から大きく変化するため，要素の消滅とか要素形状がいびつになるなどの問題が生じる[8)]ので，要素の更新も大きな課題である。図8は3次元モデルによって片持ちばりの形状を最適化したもので，(a)ははりの高さ方向にのみ形状修正していったものであり式(14)に対応している。(b)ははりの幅方向に形状修正したもので，3次元物体としてとらえることによってより自由な形状最適化が行えることを示している。図9は(a)のような継手が穴の部分で上向きに負荷されたときの形状決定例で，(d)は横断面形状，(e)は縦断面形状である。図10は(a)のように2方向から引っ張られる孔あき板の孔の部分の形状最適化例である。この例では荷重の不確定性を考慮している。(b)は $p：q=2：1$ の比率で同時に両負荷が作用するときの解，(c)は両荷重の大

(a) 靴曲面

(b) 初期形状

(c) 平面展開形状

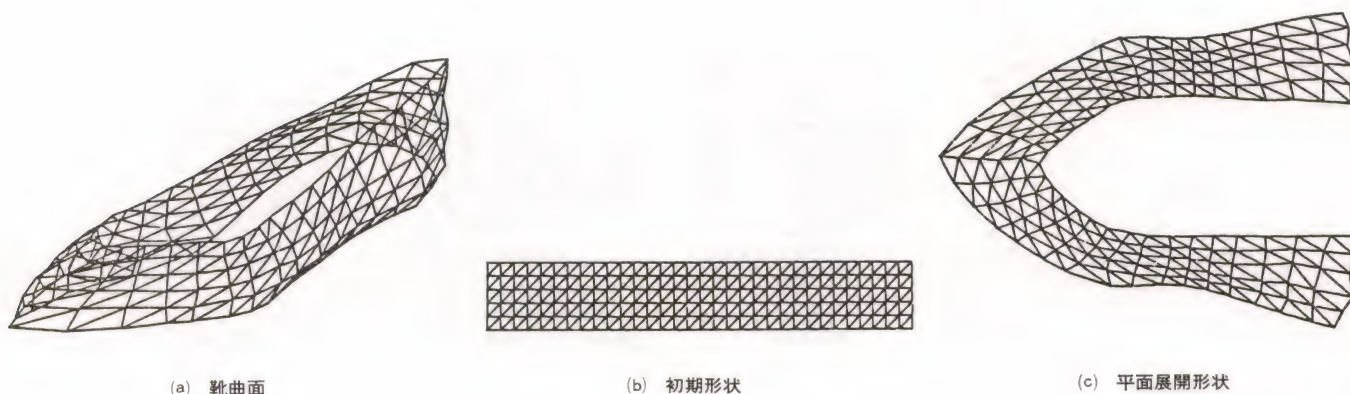



図12 靴曲面の平面展開

きさの比率は $p：q=2：1$ であるが，1/2 ずつの確率でどちらか一方だけが作用すると仮定したときの解，そして(d)は荷重の大きさの平均値の比率は $\overline{p}：\overline{q}=2：1$ であるが，それぞれその大きさが独立に変動すると仮定したときの解である。図11は，(a)のような構造物の頭部に作用する荷重の位置が45°～90°の範囲で一様の確率で変動するときの最適化の結果である。

ここで示したものは簡単な例ではあるが，形状設計変数による形状最適化によって形状最適化設計としての自由度の高い取扱いが可能となることがわかるであろう。その結果得られた形状は平面的・空間的に自由な曲線・曲面となり，どこか美しさを感じさせる。

構造最適化・形状最適化法の具体的な方法の記述は他に譲るが[16]，計算工学とともにより実用的な計算が進められていくと思われる。また，このような自由曲面は従来，加工上困難であったが，数値制御工作機械，多軸の加工機その他の進歩によって実製作が可能となってきたので，CAE, CAD の中にも形状最適化設計がサブシステムとして取り入れられるであろう。

なお，形状最適化としては，ここであげたものの他に骨組み構造物の位相，および節点位置の最適化[17]，連続体の連結関係の決定などもあるがこれについては省略する。

自由曲面の設計のための基礎技術

CAD, CAM, CAE システムにおいて，3次元物体を設計，モデリング，ハンドリングするためにはさまざまな技術が必要とされる。グラフィックスとしてのサーフィス・モデル，ソリッド・モデル，シェーディングなどの表示機能は，本誌で取り上げられているように，近年盛んに研究が進められている。形状の最適化設計においてもこれらの機能は欠かせない。また，有限要素メッシュ・節点の自動発生と，前述のような有限要素メッシュの自動更新，スプラインなどを利用するスムージングも自動的に行われるべきである。

これらとは別に，自由曲面を扱う分野の一つにその平面展開問題がある。靴や被服の設計においては，原型設定のための平面展開が必要である。この平面展開形状を求める問題は一種の形状最適化問題と考えられる[18]。また，この平面展開手法は，自由曲面をハンドリングするための，例えば自由曲面の加工のための曲面ロケータ[19]として応用することができる。**図12**(c)は，(a)の靴曲面の平面展開形状を(b)を初期値として求めた結果である。

おわりに

形状最適化は，ニーズの多様化とともにますます需要が増してくると思われる。非線形構造解析，非定常解析，動的解析の必要な分野をはじめとして2次元・3次元の構造解析，感度解析と，要素分割の自動化など形状最適化設計の発展にはこれからも計算力学に負うところが大きい。

参考文献

1) 神戸新聞：「なぜかブーム流線形」，1989.2.11
2) J.N.Siddall :"Analytical Decision-Making in Engineering Design", Prentice-Hall, p. 1 1972
3) A.Л.ЕГОРОВ(豊田訳)：「強さとかたち＝材料力学入門＝」，東京図書，p.64,1963
4) D.W. Thompson (柳田，遠藤，古沢，松山，高木訳)：「生物のかたち」，東京大学出版会，p. 135，1973
5) R.H.Gallagher, O.C. Zienkiewicz (Ed.) (川井，戸川監訳)：「最適構造設計—基礎と応用—」，培風館，1977
6) E. Atrek et al. (Ed.)："New Directions in Optimum Structural Design"， John Wiley & Sons，1984
7) G.I.N. Rozvany, B.L. Karihaloo (Ed.)："Structural Optimization", Proc. of IUTAM Sympo. on Structural Optimization，Kluwer Academic Publishers，1988
8) R.T. Haftka, R.V. Grandhi ："Compt. Meth. in Appl. Mech. Engng.，57，p.91 1986
9) 室津他：機械学会論文集，46-404， A,p. 420，1980.4
10) 尾田，山崎：機械学会論文集，43-368，p. 1182，1977.4
11) 梅谷，平井：機械学会論文集，42-364，p. 3754，1976.12
12) 瀬口，多田：機械学会論文集，44-381，p. 1469，1978.5
13) 多田，瀬口，藪：機械学会論文集，51-471，A,P.2536, 1985.11
14) 多田，竹谷：機械学会論文集，53-487，A,p.667，1987.3
15) Y.Tada, Y.Seguchi (G.Yagawa, S.N.Atluri, Ed.):"Computational Mechanics'86, Theory and Applications", Proc. of Int'l Conf. on Compt.Mech., Vol.2, X-39, Springer-Verlag, 1986
16) 尾田他(日本機械学会編)：「構造・材料の最適設計」，技報堂，1989(予定)
17) 尾田：機械の研究，40-12，p.1361，1988.12
18) 島田，多田他：機械学会論文集，54-498，C，p.497，1988.2
19) 島田，多田：情報処理学会論文誌，29-5，p.539，1988.5

期待される地図情報処理

地籍情報管理システム ランドマン

このシステムは，国土庁標準フォーマットによる国土調査成果のデータ作成とそのデータベースをもとに，市町村の土地行政業務の合理化に照準をあてた。

城岡　優*

はじめに

地籍調査とは，「国土の開発及び保全並びにその利用の高度化に資するとともに，あわせて地籍の明確化を図る」という国土調査法の目的に従い，国有林，水面湖沼などを除く全国土にわたって統一的な調査をすることを目標に，国庫補助事業として昭和26年度から実施されている。地籍調査は，一筆ごとの土地についてその特徴・実態を明らかにするため，所在，地番，地目，境界の調査と登記簿に記載された所有者に関する確認と境界の測量および面積の測定を行い，調査の結果を地図および簿冊に作成することで，いわば土地に関する戸籍調査ともいうべき基礎的な調査である。

◉図1，2は次ページ。

今日の測量技術，情報処理技術などの進展に従い，国土庁を中心とした国土調査問題研究会（昭和57～59年度），国土調査問題専門検討会（昭和60～63年度）などの提言により（**図1**），昭和61年地籍調査作業規程準則および同運用基準が改正され，「地籍調査成果のシステム化の実施について」の指示があり（**図2，表1，2**），市町村が保管する地籍図および地籍簿を用いて調査成果のシステム化が活発になってきている。システム化の内

表1　標準的記録形式のファイル仕様

項　目	仕　様
ファイル編成	単ファイル
レコード長	固定長（128バイト／レコード）
コード	磁気テープの場合：EBCDIS，ただし漢字はJIS 磁気フロッピディスクの場合：ASCII，ただし漢字はJIS
その他	磁気テープ：ノンラベル，ブロッキング可 （ブロッキングファクタは任意） 磁気フロッピディスク：IBM形式

表2　標準的記録形式のファイル内容

レコード	内容など
共通情報	市町村名，市町村コードなど 字名、字コードなど 図葉コード，分割図葉など
筆属性情報	当該格納単位区域が含まれる字の字コード， 調査年月など 各筆の地番，地目，地積，所有者情報など
筆界未定情報	互いに筆界未定となっている筆の地番の表
筆図形情報	測量方法，数値変換方法，精度，調査年月など 各筆の筆界点の位置座標の表
筆界点情報	筆界点番号と位置座標の対応表 （筆界点番号が付されている場合のみ）
図根点等情報	図根点などの番号，種別，位置座標の対応表
その他	トレイラー・レコード （ファイルの最後を示す）

*しろおか　ゆう　　桜井㈱　マーケティング部　〒110　東京都台東区東上野1-11-12

(1) 国土調査問題研究会最終報告(昭和60年3月)

(2) 国土調査問題専門検討会

60年度：パソコンを用いた地籍調査情報システムの普及と高度化への対応

↓

地籍調査に関係する地図情報の数値化技術および地理的情報システムの現状調査

↓

データ記録形式等の標準化案の提言

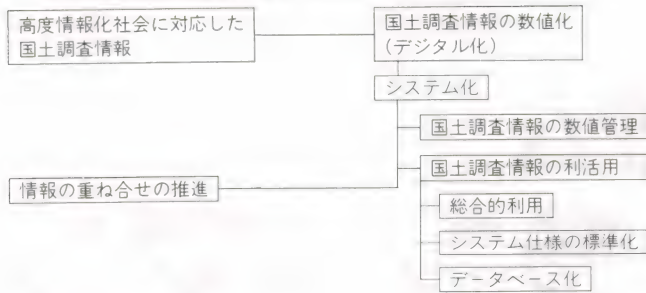
↓

61年度：すでにシステムを導入している市町村へのアンケート調査

↓

地籍調査情報システムの標準的仕様案およびシステムの利活用の方向に関する提言

図 1

図3 地籍調査の作業手順とシステムの基本的機能

(1) 地籍集成図の作成

地籍集成図は，隣接する複数の地籍図などの写しを原則として次に示す規格に集成編集し，作成するものとする。

●国土基本図 ●都市計画図 ●国有林の基本図 ●森林計画図,森林施業図または森林基本図 ●その他の地形図——などより選定し,地籍集成図の図郭の寸法は縦60cm，横80cmとする。

(2) 数値情報化の実施

数値情報化は，地籍図などの写しを原則として次に示す方法および規格により，数値情報として記録するものとする。

(a) 数値法または航測数値法により筆界点などの平面直角座標がある場合には，当該座標値を利用するものとする。

(b) (a)以外の場合には，地籍図等の写しを数値変換して得られた平面直角座標値を利用するものとする。

(c) 数値情報の記録媒体および記録形式は表1,2などより，記録形式の標準化（国土庁標準フォーマット）から,各システムでそれぞれの形態で作成・保管されている地籍情報を相互利用できる。

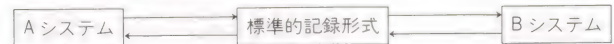



図 2

容は，地籍集成図の作成と数値情報化の実施である。このシステム化の標準的システムのハードウエアは，カラー・ディスプレイ，パーソナルコンピュータ本体，ハードディスク，プリンタ，8インチFDD，X-Yプロッタ，デジタイザなどにより構成されており，ソフトウエアとの組合せにより，地籍調査支援システム，数値情報化システム，地籍情報管理システム，地籍情報利活用システムなどに分類される（**図3**)。

これらのシステムは地籍調査の作業手順により，地籍調査事業工程におけるA工程からH工程まで網羅されている。各作業工程内容の概要は次の通りである。

①A工程

地籍調査事業主体における事業計画の策定およびこれに伴う事務手続き。

②B工程

地籍調査事業実施主体における事業着手のための準備。

③C工程（地籍図根三角測量）

地上法による地籍測量において，所定の密度で配置された図根点（地籍図根三角点）を設置し，その位置を基本三角点，四等三角点などを基礎として測量する作業。

④D工程（地籍図根多角測量）

地上法による地籍測量において，所定の（中程度）密度で配置された図根点（地籍図根多角点）を設置し，地籍図根三角点などを基礎として測量する作業。

⑤E工程（一筆地調査）

土地利用の現況を把握するため，土地登記簿および字限図の写しをもとにして現地において関係土地所有者立ち会いのもとに，毎筆の土地についてその所有者，地番，地目および境界に関する調査で地籍簿作成の基礎となる調査。

⑥F工程（地籍細部測量）

地上法による地籍測量において，地籍図根多角点などを基礎として各筆の筆界を測量し，地籍図原図を作成する作業で細部図根測量(前半)と一筆地測量(後半)からなる。

・ランドマン「ジム」システム(地籍調査支援システム):①~⑥
・ランドマンシステム(地籍情報管理システム):①~⑧

図4　システム構成

⑦ G工程（地積測定）

地籍細部測量により求めた筆界点の座標値または作成された地籍図原図をもとに，毎筆の土地の面積を計算または測定する作業。

⑧ H工程（地籍図および地籍簿の作成）

一筆地調査および地積測定の結果に基づき地籍簿案を作成し，この地籍簿案および一筆地測量により作成された原図を20日間一般の閲覧に供して，成果品たる地籍簿および地簿図を作成する作業。

以上の工程により，国・都道府県の行政指導のもとで市町村が事業主体となって実施し，市町村が行う業務のうち地籍測量，地籍図の作成，地積測定作業のほとんどは専門の作業機関（測量業者）に発注依頼して実施されているので，市町村が直接行う作業としてはそれらの発注や検収と一筆地調査，地籍簿の作成などが主なものである（一部の市町村において，全工程を直営で実施している所もある）。

ここに紹介する地籍情報管理システム「ランドマン」は全工程を網羅し，事業計画から調査票・調査図・文書作成処理，測量計算・集計業務，認証業務，登記業務まで，地籍情報のすべてをコンピュータが一貫処理するシステムである。

システムの基本構成

基本的なハードウエアは，PS 5570，16インチ・カラーディスプレイ，キーボード，プリンタ，8インチFDD，カセット・ストリーマ，タブレット，X-Yプロッタからなる。主なソフトウエアは，事業計画処理，調査票処理，調査図処理，文書作成処理，基準点処理，情報登録更新，測量計算，分筆・合筆処理，地積計算処理，集計計算処理，地籍図・一筆図・分筆図・字図・地積測量図・地籍集成図などの作成処理，地籍簿作成処理などからなる（**図4**）。

入力情報・処理・出力情報

操作については，ディスプレイに表示される内容によりテンキーまたはカーソルで入力する。地名・人名などについてはワードプロセッサと同様の入力方法である。操作手順は今までの手作業と同様であり，以下に操作方法と処理内容を示す。

◆ 国土庁標準フォーマット地籍情報管理システム ◆ LANDMAN IBM-5570

1 : 地籍情報管理システム
2 : 地籍予備登録システム
3 : 8インチからのデータ変換
4 : 作業地区変更
5 : 作業地区データ初期化
+ : 作業の終了

【 番号または↑↓←→で選択して下さい 】
処理番号を入力して下さい →→
CAPS 半角

図 5

◆ 処理選択画面（メインメニュー）◆ 地籍情報管理システム ver1.0

1 : 事業計画処理	8 : 集計計算処理
2 : 調査票処理	9 : 調査票・地籍簿作成処理
3 : 文書作成処理	10 :
4 : 基準点成果簿	11 :
5 : 情報登録更新	12 :
6 : 分筆・合筆処理	13 :
7 : 地積計算処理	+ : 作業終了

【 番号または↑↓←→で選択して下さい 】
処理番号を入力して下さい →→
CAPS 半角

図 6

◆ 予備処理 (ﾒｲﾝﾒﾆｭｰ) ◆　　地籍情報管理システム Ver1.0

1：字名登録　　6：
2：住所定型句登録　　7：
3：所有者登録　　8：
4：図版登録　　9：
5：地目登録　　+：予備処理の終了

【 番号または↑↓←→で選択して下さい 終了=■ 】
処理番号を入力して下さい →→

CAPS 半角

図 7

(1) ランドマン「ジム」(地籍調査支援システム)

①A・B工程である事業計画・準備処理により，事業全体の事業費の算定，事業計画地区別説明調書などの作成（**図5，6**；次ページ)。

②地籍予備登録処理により，調査地区の字名・住所定型句・所有者・図版・地目などの情報を事前に入力する（**図7**)。

③E工程である調査票の処理は②の予備登録データから引用し，ディスプレイ上の補助画面により選択し，土地登記簿，固定資産税台帳より入力して地籍調査前調査票を作成する（**図8**)。

④文書作成処理により③のデータを引用し，名寄簿・立会通知書・立会確認書・所有者名簿・所有者一覧表・宛名ラベルな

◆ 地籍調査票処理 ◆ 修 正　　地籍簿作成システム ver1.0
【データ番号 1 所在　　地番 50 】

番号	コード	所有者名
1	12345601	桜井 太郎
2	12345602	桜井 次郎
3	12345603	星 光
4	12345604	清水町
5	12345605	田中 花子

4 所有者	登記簿住所	[大字清水字美保５０] [ ]	
	氏名又は名称	1 (12345601) 桜井 太郎	
5	登記関係表示事項	所有権 既登記	その他の登記 有無 無

次ｺｰﾄﾞ参照 Pf9
前ｺｰﾄﾞ参照 Pf8

(未登記=■ 新規登録=■ ｺｰﾄﾞ入力=■ 表示NO.変更=■)
所有者番号を入力して下さい →→

CAPS 半角

図 8

◆ 処理選択画面 (ﾒｲﾝﾒﾆｭｰ) ◆　　地籍情報管理システム ver1.0

1	事業計画処理	8	分筆・合筆処理
2	調査票処理	9	地積計算処理
3	調査図処理	10	集計計算処理
4	文書作成処理	11	三角点・多角点網図処理
5	基準点成果簿	12	調査票・地籍簿作成処理
6	情報登録更新	13	地籍図・筆界点番号図 地籍図字一覧図作成処理
7	測量計算	+	作業終了

【 番号または↑↓←→で選択して下さい 】
処理番号を入力して下さい →→ 4

英数 半角

図 9

```
*** 地籍調査異動項目データリスト ***

:( 登録番号 )  1                                   ( 調査図番号 ) 1
:( 所　　在 )  2) 大字上野字湯島                   ( 地　　番 ) 24-1
:( 地　　目 )  7) 山 林                            ( 地　　積 )    1018.00
:( コ ー ド ) 8001405       ( 郵便番号 ) 011       ( 氏　　名 ) 桜井 太郎
:( 現住所   ) 栃木県宇都宮市飯田町９５－１７
:( 登記住所 ) 栃木県宇都宮市飯田町９５－１７                          (        )
:( 所有権登記 ) 既登記                              ( その他登記 )
:( 権利項目 ) 1)        2)        3)               4)        5)        6)
:( 地目変更 ) 昭和 年 月 日                        ( 不詳地目変更 )
:( に 分 割 )        ～        1)     2)     3)    4)     5)
:( から分割 )      ( に合併 )      ( に一部合併 )  ( 地番変更 ) 24
:( を 合 併 ) 1) 24-2     2)      3)               4)        5)
:( を一部合併 ) 1)        2)      3)               4)        5)
:( 所在変更 ) 昭和 年 月 日                        ( 表示登記 ) 昭和 年 月 日
:( 滅　　失 ) 昭和 年 月 日                        ( 訂　　正 )
:( 筆界未定 ) 1)          2)      3)               4)        5)
:             6)          7)      8)               9)        10)
:( 地積訂正 )             ( 不存在 )               ( 現地確認不能 )

<< 分筆・合筆データチェックリスト >>

[NO.] [ 字　　名 ] [ 地　番 ] [ 地　目 ]
      [ 所　有　者 ]                    ] [ その他登記 ]

【 合 筆 前 】

[ 1 ] [大字上野字湯島 ] [24 ] [山 林 ]
      [桜井 太郎                        ] [ なし ]

【 合 筆 後 】

      [大字上野字湯島 ] [24 ] [山 林 ]
      [桜井 太郎                        ]

*** 地籍調査後データリスト ***

:( 登録番号 )  1
:( 地籍図番号 ) U13-3
:( 所　　在 )  2) 大字上野字湯島                   ( 地　　番 )  24
:( 地　　目 )  7) 山 林                            ( 地　　積 )     1604.00
:( コ ー ド ) 8001405       ( 郵便番号 ) 011       ( 氏　　名 )  桜井 太郎
:( 現住所   ) 栃木県宇都宮市飯田町９５－１７
:( 住所変更 ) 1)                 2)
:( 所有権登記 ) 既登記
```

図 10

どの作成をする（**図 9**）。

⑤情報登録更新により現地調査結果を異動事項・共有者名簿・分合筆チェックなどに入力し，調査後データ登録・地番データ並べ換え・調査前後データチェックなどを行う（**図 10**）。

⑥8インチ・フロッピディスクからのデータ変換処理により測量業者が実施したC，D，F，G各工程のデータを引用して②～⑤のデータと合成させ，地籍調査後の調査票，各種集計表（図版別，地目別，字別，所有者別）などの出力をし，地籍簿案・地籍図案（測量業者よりの出図）などを作成して閲覧を行い，修正などがあればデータを修正し処理をする（**図 11**）。

⑨調査票・地籍簿作成処理により地籍調査票・地籍簿を出力し，

◆　処理選択画面（メインメニュー）◆　　地籍情報管理システム　ver 1.0

1 ： 8インチからのデータ変換	9 ： 字名データリスト
2 ： ダンプリスト作成	10 ： 図葉データリスト
3 ： 市町村・字名・図葉レコード	11 ： 所有者データリスト
4 ： 所有者情報変換	12 ： 図根点・筆界点座標リスト
5 ： 筆界点・図根点　座標変換	13 ： 筆界点データリスト
6 ： 面積データ変換	14 ： 筆界未定データリスト
7 ： 筆界未定レコード変換	15 ：
8 ： 市町村データリスト	＋ ： 作業終了

【 番号または↑↓←→で選択して下さい 】

処理番号を入力して下さい →→

CAPS 半角

図 11

地籍調査前の土地の表示							地籍調査後の土地の表示								
字名	地番	地目	地積 ha	a	㎡	所有者の住所及び氏名又は名称	字名	地番	地目	地積 ha	a	㎡	所有者の住所及び氏名又は名称	原因及びその日付	地図番号
大字上野字東上野	113	畑		6	34	上野郡大字上野３２２９ 桜井　三郎			山林		5	20		年月日不詳地目変更 地積錯誤	U23-1
	114-イ	原野		3	30	上野郡大字上野３５１９ 桜井　四郎		114	山林		6	19		年月日不詳地目変更 114と地番変更 地積錯誤	U23-1
	114-ロ	原野		3	30	上野郡大字上野３１９６ 三河屋　保								年月日不詳地目変更 116に合筆	
	115-ロ	山林		2	08	上野郡大字上野４７１－２ 高木　京子		115			1	28		115と地番変更 地積錯誤	U23-1
	115-1	山林		3	90	上野郡大字上野３１９６ 三河屋　保								116に合筆	
	116	畑		8	19	同　　上			山林					年月日不詳地目変更 115-1，114-ロを合筆	U23-1
	117-イ	畑		9	28	栃木県宇都宮市飯田町９５－１７ 桜井　太郎		117-1	山林		8	76		年月日不詳地目変更 117-1と地番変更 地積錯誤	U23-1
	117-ロ	畑		15	27	同　　上		117-2	山林		14	39		年月日不詳地目変更 117-2と地番変更 地積錯誤	U23-1
	118-1	山林		5	58	上野郡大字上野３１７５ 三河屋　巌					7	41		地積錯誤	U23-1
	118-2	畑		6	54	同　　上					6	67		地積錯誤	U22-2

```
    <<   図 根 点 等 情 報     >>
9  図根点等レコード
                                                                                                    PAGE    1
:---------1---------2---------3---------4---------5---------6---------7---------8---------9--------10--------11--------12--------:
:842-U22-2   +067071718-03916828143-U22-2   +067076937-03916312844-U22-2   +067081898-039160390
:845-U22-2   +067085789-03916858946-U22-2   +067091640-03917160147-U22-2   +067108750-039177441
:848-U22-2   +067107281-03916917150-U22-2   +067138382-03912707051-U22-2   +067126593-039132550
:852-U22-2   +067109593-03914435953-U22-2   +067106867-03914401954-U22-2   +067102640-039143261
:855-U22-2   +067089937-03913855056-U22-2   +067200000-03913023057-U22-2   +067196796-039129621
:858-U22-2   +067200000-03913864859-U22-2   +067199023-03913780060-U22-2   +067200000-039137109
```

図 13

◆　測　量　計　算　（メインメニュー）◆　　　　地籍情報管理システム ver1.0

１	計算条件設定	誤差配布等の諸条件を決定
２	座標登録・修正・抹消	使用する座標の登録等を行う
３	登録座標リスト	登録されている座標の一覧表作成
４	多角測量計算	路線多角（ 結合・閉合 ）の計算
５	視準角によるトラバース計算	オープントラバース計算
６	交点計算	１２種の交点計算
７	逆座標計算	座標差による距離と方向角の計算
８	座標面積計算	経線距法又は倍横距法による計算
９	点のプロット	プリンターによる点のプロット
１０	簡易ＨＡＯ網平均計算	条件方程式により交点を計算
１１	簡易ＸＹ網平均計算	３，４路線の重量平均より交点を計算
１２	点検座標計算簿作成	網平均計算の点検座標計算簿を作成
１３	精度管理表作成	三角測量・多角測量の精度管理表作成
１４	基準点抄写簿作成	図版内の基準点のリストを作成
１５	次数別路線名目次作成	次数別に路線名の目次を作成
＋	作業終了	

処理番号を入力して下さい →→→　　【 番号または↑↓で選択して下さい 終了=■ 】

英数 半角

図　14

◉ 図 13 は前ページ。

認証準備をする（**図 12，13**）。

⑩ 8インチ・フロッピディスクからの変換処理により測量業者から納品されたデータをダンプリストとして出力させ，土地の異動などに対応する（図 13）。

(2) ランドマンシステム（地籍情報管理システム）

このシステムは，ランドマン「ジム」の処理内容に測量業者が実施する調査図処理，測量計算，地積計算処理，三角点網図，多角点網図，地籍図，筆界点番号図，地籍図字一覧図，地積測量図，地籍集成図，8インチ・レコードファイル作成などの処理を行う（**図 14**）。

①調査図処理により，公図または字限図をタブレットで読み取り，調査地区をX-Yプロッタで出図処理をする。

②測量計算処理では，地籍測量による成果品の内訳を各工程別で表すと，C工程では基準点等成果簿写・地籍図根三角点網図・地籍図根三角点成果簿・地簿図根三角測量精度管理表，D工程では地籍図根多角測量観測計算諸簿・地籍図根多角点網図・地籍図根多角点成果簿・地籍図根多角測量精度管理表，E工程では調査図，F工程では細部図根測量観測計算諸簿・数値図根点配置図・数値図根点成果簿・一筆地測量観測計算諸簿・筆界点番号図・筆界点成果簿・地籍細部測量精度管理表・地籍図素図，G工程では地積測定観測計算諸簿・地積測定成果簿・地積測定精度管理表・地目別筆数面積集計表，H工程では地籍図一覧図・地籍図原図・地籍明細図——などを出力する（**図 15**）。なお，C工程の計算プログラムはNTTデータ通信のプログラムを使用して処理を行う。

③ 8インチ・レコードファイル作成処理により，筆界点座標値などの磁気記録をフロッピディスクに書き込む。

以上でランドマンシステムの説明を終わりにしたい。

図　15

地　籍　調　査　票

調査図番号	1	地籍図番号	U13-3
地籍調査前の土地の表示		地籍調査後の土地の表示	
所在・地番	大字上野字湯島　24番1		大字上野字湯島　24番
地目・地積	地目　山　林　地積　1018 ㎡	地目	地積　1604 ㎡
所有者　住　所	栃木県宇都宮市飯田町95－17		
所有者　氏名又は名称	桜井　太郎		
登記関係	所有権　その他の登記	地籍簿へ記載の事項	24-2を合筆 24と地番変更
表示事項	㊐・未　無し		
異動事項	○ 年月日不詳地目変更 ○ 昭和　年　月　日（　）地目変更 ○ に分割 ○ から分割 ㋹ 24-2 を合併 ○ に合併 ○ 番に一部合併　番の一部を合併 ㋹ 24 番と地番変更 ○ 昭和　年　月　日　と所在変更 ○ 昭和　年　月　日　新たに表示登記をする土地 ○ 昭和　年　月　日（　）滅失 ○ を　と　訂正 ○ 番との筆界未定 ○ 地積訂正　○ 不存在　○ 現地確認不能	所有者意見	左記のとおり分割・合併・一部合併・滅失・不存在・地番変更・　について同意（承認）をする。 昭和　年　月　日 土地所有者（代理人）　㊞
		備考	現　地　調　査　昭和　年　月　日 立会人氏名

地籍図番号 U13-3,....
24-2を合筆
24と地番変更

◀図 12

ランドマンシステム評価と今後の展望

このシステムは，地籍情報のすべてをコンピュータに数値化して保存・管理し，国土庁標準フォーマットによる地籍調査成果のデータ作成だけでなく，そのデータを市町村の土地行政業務の合理化・住民サービス・登記業務など周辺の業務にも活用できるものである。155 M バイトのデータが1巻のカセットに格納できる大容量設計の採用で，カセット交換により都市部の複数地区の処理も簡単に行えるようにした。

今後の展望として，現在のシステムにスキャナやカラー・プリンタなどの増設により，固定資産税務，消防・防災，上下水道管理などのプログラムを開発し，市町村役場における無理のないシステムアップを行っていきたい。

多様情報を統合するハイパーメディア概論

多様情報統合処理のためには，情報自体の表現モデルとそのユーザー・インタフェースが必要である。本稿では後者の有望株「ハイパーメディア」について論じる。

白田　由香利*
國井　利恭**

★はじめに

現代社会には文字，グラフィックス，音，ビデオなどの多様な情報があふれている。それらをコンピュータで扱っていく場合，多様なままウィンドにはめ込んでしまう雑居型のマルチメディアがある。しかし，これでは多様な情報から必要なオブジェクトやその集まり，あるいはオブジェクトの一部を目的に合わせてある順序で指定したり，引っ張り出したりして変更する情報利用の効率化が困難である。この困難を乗り越えるには，まずこれらのオブジェクトを統合的に表現できるモデルが必要である。次に，このモデルを拡張してモデル間の関係付けや，モデルのサブモデルへの分解などができるようにする必要がある。このような拡張モデルが，データベース管理システム分野では，ポスト・リレーショナル・モデルとしてグラフ・データモデル[1]により，また実用システムではこのモデルを実現したG-BASE[2]として利用可能となった。拡張モデルとは並列に，この多様な情報の統合処理のためのユーザー・インタフェースとして有望視されているものに，ハイパーテキストあるいはハイパーメディアとよばれる電子文書形態がある。現在，両者が統合されるにはいたっていない。拡張モデルについては稿を改めていずれ紹介することにして，本稿では取りあえず統合型ユーザー・インタフェースから紹介することにする。

はじめにハイパーメディアとはどのようなものかを述べ，次に実際に広く世の中に普及しているマッキントッシュのハイパーカード（HyperCard）の基本的な特徴について述べる。その後はハイパーメディアの今後解決していくべき問題点について考察し，そのブレークスルーを探っていきたい。

★ハイパーメディアとは

まずはじめに，従来から行われてきた一次元的情報処理法を考察してみよう。

例えばカード整理法がある。この方法で人はアイデアやメモなどをカードに記録しておき，ある程度の量のカードが蓄積された時点でそれらを並べてカードの関連付けを試みる。そのとき，同じ部類に属するカードや因果関係で結ばれているカードなどを近くに配置してみたり，同じマークをカードに書き込んだりしてみる。しかし，壁に貼り付けた多数のカードをそのまま保存しておくわけにもいかないので，カードははずされて集められ，一次元的にまとめてカードラックなどにしまわれる。カード情報間の関連情報がこの時点でほとんど消えてしまう点が問題である。

書籍も人にとって非常に有効な情報源である。書籍は紙に情報が印刷され，一次元的に並べて綴じてあるメディアである。人が書籍をひもとくとき，関連事項の頁を調べてそこをめくる。マーキングのためにしおりなどをはさむかもしれない。関連項目に関する情報は本文中に書いてあったり，また Knuth の本[3]のように，本の巻頭でシリーズ本を読む順序についてのフローチャートを載せてあるよ

*しろた　ゆかり（博士課程大学院生）　**くにい　としやす　東京大学　理学部情報科学科　〒113　東京都文京区本郷7-3-1

うなものもある。

辞書の場合，関連を明示するための工夫はより顕著である[4]。例えば百科事典である項目を調べようとする場合，キーワードから目的の記述が存在する箇所をインデックスなどにより探していく。その記述中にはその関連項目が記してあるので，さらに詳しく調べていくことも可能である。

以上をまとめてみると，従来のメディアでは情報は一次元的に配列されているだけで，その間の関連を表現する情報（以後これをリンク情報とよぶことにする）は含まれていない。また含まれていたとしても本来の内容と同じレベルで記述してあるだけなので，関連項目を調べるときは人がそれを読み取ってその場所を探し出さなくてはならなかった。具体的にはカードを見つけたり，本の頁をめくったりする行為がこれに相当する。

このリンク情報の処理をコンピュータに肩代りさせてくれるものがハイパーメディアである。例えば，CD-ROMの百科事典をハイパーメディアを使って読んでいたとしよう。ハイパーメディアの画面中に「遠赤外線」という見慣れない言葉がでてきたので，それについて詳しく知りたいと思ったとする。ユーザーは「遠赤外線」という文字列の部分をマウスでクリックすると，瞬時に「遠赤外線」を説明しているカードがスクリーン表示される。

ハイパーメディアではリンクという概念が重要である。ハイパーメディアの内部では，ユーザーが知りたい/扱いたい本来の情報とその間のリンク情報を分離して保存していて，ユーザーがリンクをたどりたいと指示したとき，ハイパーメディアはそのリンク情報によってターゲットであるカード情報にコントロールを移動させる。

ここで，ハイパーメディアとは何であるかを以下にまとめる。

- 情報はカードあるいはフレーム，ノードなどとよばれるユニットとしてまとめられている。
- 情報ユニットに含まれる情報の型としては，(1)テキスト，(2)グラフィックス，(3)サウンド，(4)アニメーション，(5)ビデオなどがある。
- 1つの情報ユニットの表示は1つの画面に対応する。マルチウィンドウ・システムの場合，1つのウィンドが対応する。
- 情報ユニットはリンクによって関連付けられる。
- ユーザーは情報ユニットを作成／編集し，それからユニット間にリンクを作成する。
- ユーザーはリンクをたどることにより，情報の検索を行う。
- ユーザーはリンクをたどることにより，欲しい情報だけを選択的に入手することが可能になる。
- 情報ユニットは一次元的ではない。ユーザーがリンクを作成/編集することにより，ユーザーの要求に合う情報ストラクチャを形成することができる。この構造はネットワーク型，またはグラフなどとよばれている（両方ともハイパーメディアに限らない一般用語である）。図1にネットワーク構造の例を示す。

★マッキントッシュのハイパーカード

■ハイパーカードの概要[5),6)]

ハイパーカードは1987年8月11日にボストンで開かれた“Macworld Expo”で，アップル・コンピュータ社により発表された。このときのビデオディスク・プレーヤーを使用したデモが印象的だったこともあり，「プログラミング知識のないユーザーでも自由にアプリケーションを作成できる」といううたい文句とともに，ハイパーカードはマッキントッシュ・ユーザーの間に大きな衝撃となって広まった。この開発チームのリーダーを務めたのは，マックペイントの作者であるBill Atkinsonであり，この開発には3年を要した。

ハイパーメディアのアイデア自体は，なんと1945年にVannevar Bushによる“As We May Think”という論文に発表されている。そのMemexと命名された彼のシステムでは，リンクをたどることによりマイクロフィルムが次々表示できるようになっている。当時の技術ではMemexは実現できなかったが，これはまさにハイパーメディアのルーツといえよう。

現在，ハイパーカードはマッキントッシュのバンドルソフト（マッキントッシュを購入すると無料で付いてくるソフトウエアをこのようによぶ）になっている。マッキントッシュのユーザーは全世界にわたっており非常に多いが，そのユーザーたちがハイパーカードでカードやスタックを作成するので，ハイパーメディアの情報蓄積量ではハイパーカードが最大である。アップル・コンピュータ社としても，ハイパーカードをマッキントッシュ上の情報の標準フォーマットとすることにより，ユーザー間の情報の共有化な

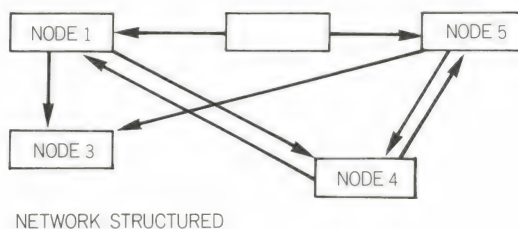



図1　ハイパーメディアの構造は(1)ノード（情報ユニット）とその間を結ぶ(2)リンクから構成される。

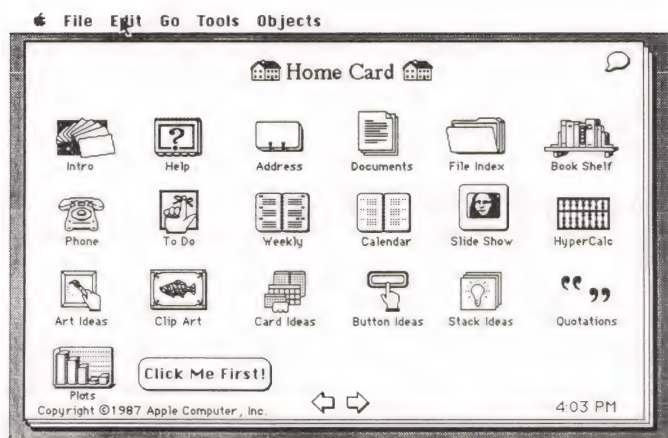



図2 ハイパーカードを起動すると，ホームスタックの先頭にあるこのホームカードが表示される。

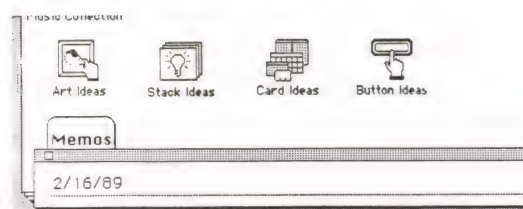



図3 メッセージボックスを通してユーザーは，対話的にハイパートークのファンクションを実行できる。

らびにアプリケーションの共有化が図れるというメリットがある。まさにハイパーカードは，マッキントッシュの核をなしているといえる。

扱える情報の型は，(1)テキスト，(2)グラフィックス，(3)サウンド，(4)アニメーション，(5)ビデオ（ビデオディスク・プレーヤーなどの駆動機能が提供されているのみで，同じスクリーン上の1ウィンドにビデオ情報が表示されるわけではない）――などがある。音や絵を取り込むI/Oとして安価なものがそろっていることも，ユーザーにとって大きな魅力である。

しかし，ハイパーカードはあくまで個人情報処理用ハイパーメディアであり，分散環境における複数のユーザーの使用には向いていない。バージョン管理機能なども残念ながらまだない。

ハイパーカードには，ハイパートークとよばれるプログラミング言語が内蔵されている。また，マックペイントのようなグラフィック・ツールも内蔵されている。これはツールパレットの中に入っていて，メニューから選択してきてそのままスクリーン上の任意の場所に置いておける（図6）。

ハイパーカードの用語は他のシステムと少し変わったネーミングがなされているので，少し説明しておこう。

〈ハイパーカードの用語〉

- スクリプト：ハイパートークで書かれたプログラムのこと。ハイパーカード上の各オブジェクトが自分の役割を「台本」通りに演じるという意味からこのように命名された。
- スタック：一まとまりのカードの集合

HyperCard
HomeStack
Stack　Stack　Stack
BackGround　BackGround　BackGround
Card　Card　Card
Button　Field　Field

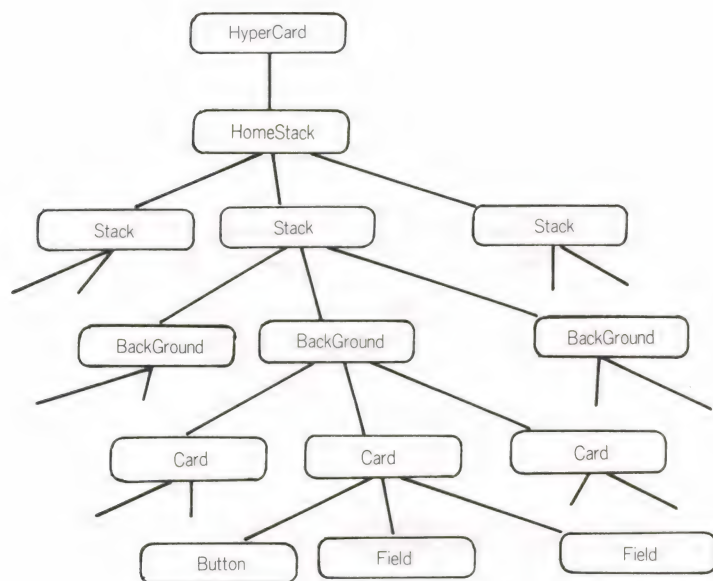



図4 ハイパーカードにおけるオブジェクトの階層構造

を指す。一次元的に順番付けられている。その中にはスクリプトも含まれるので，各スタックは1つのアプリケーションに対応する。ソフトウエアに対する言葉として「スタックウェア」がある。

●背景（バックグラウンド）：1つのスタックの中には複数のカードが含まれているが，そのうちいくつかのカードでは共通の部分がある。ハイパーカードでは，それは背景として登録できるので，各カードごとにいちいち書かなくてすむ。ボタンなども背景として登録できる。複数の背景設定もできる。

●フィールド：文字列や数値を代入できる四角いエリア。図6にあるようなスクローリング・フィールドも用意されている。

●ホームスタック：出荷時の設定ではハイパーカードを起動するとこのスタックが開かれ，その第1枚目のカード（これをホームカードとよぶ）が表示される（**図2**）。役立つツールを多く含む便利なスタックなので，ハイパーカードに慣れるまで起動時のスタックはこれにしておいた方が無難であろう。

●メッセージボックス：ユーザーがハイパーカードと対話するための特殊なウィンド。ハイパートークのコマンド（例えば“go to Home”）をここから対話的に実行できる。ハイパートークのファンクションも使用可能。例えば“the date”と入力すると，その結果が**図3**のようにメッセージボックスに表示される。四則演算のカリキュレータとしても使える。

上記のハイパーカードにおけるオブジェクトの階層化を**図4**に示す。

■ユーザーのレベル設定[6]

ハイパーカードではユーザーの要求に合わせて5つのユーザー・レベルを用意している。この設定はユーザー・プレファレンス（**図5**）で行う。このカードはホームカード（図2）の1枚前のカードなので，戻り（左向き）矢印をクリックすればよい。

以下にユーザー・レベルを列挙する。

●ブラウジング：リンクをたどりながら読むことだけは可能。カードへのテキストの書込みはできない。

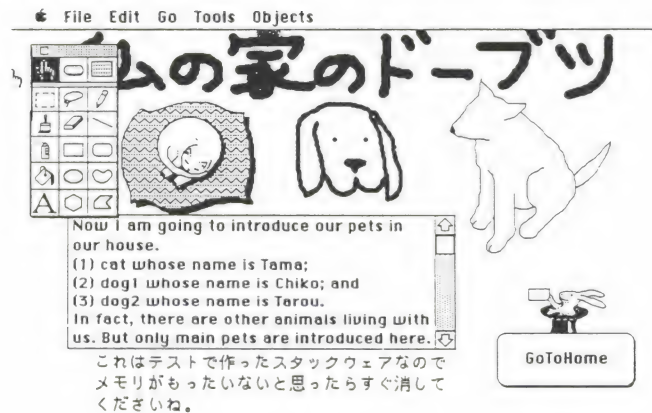



図 6

●タイピング：テキスト入力，カードの編集機能の付加。

●ペインティング：ペインティング・ツールでの描画機能の付加。

●オーサリング：カードにボタン，フィールドの付加や削除が可能になる。カードやリンクの作成ならびに変更も可能になる。コマンド選択により，対話的に目的のスクリプトが作成できるようになっている。

●スクリプティング：スクリプトを直接書けるようになる。オーサリング・レベルに比べ，ハイパートークをフルに活用できるようになる。

■オーサリング・レベルの概要[6]

では，簡単にハイパーカードのオーサリング・レベルでできることをみていこう。

File Edit Go Tools Objects
User Preferences
User Name:
User Level:
Browsing
Typing
Painting
Authoring
Scripting
Text Arrows
Power Keys
Blind Typing

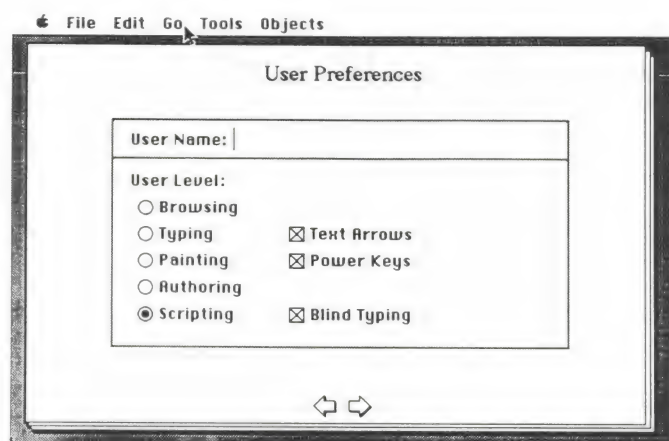



図5 ユーザー・レベルを設定するためのユーザー・プレファレンス・カード

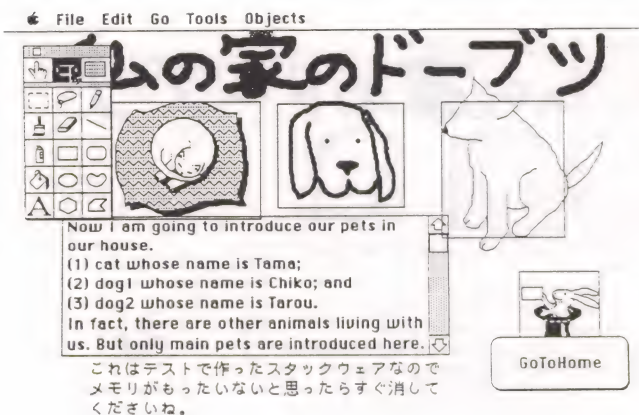



図 7

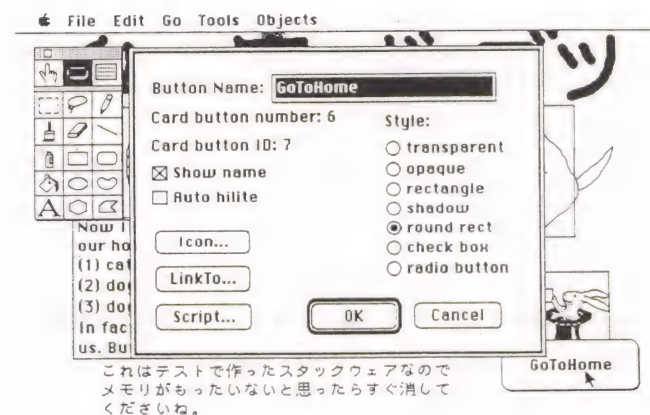



図 8

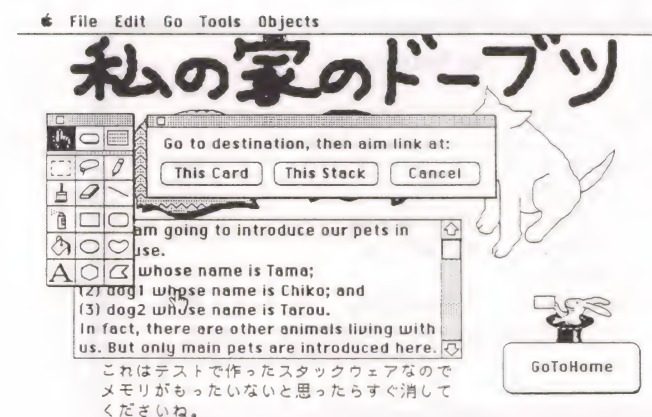



図 9

図 6 にカードが表示されている。左上にあるのがツールパレットである。通常スタックを使用するときは，この中の指マークを選択しておく。1枚のカードには普通，複数個のボタンやフィールドが含まれている。ボタンの編集をしたいときは，**図 7** のようにツールパレットの最上段真ん中のボタンマークを選択すると，ボタンエリア（ここでは猫1個，犬2個，兎1個，“Go To Home”の計5個）が現れる。動物ボタンについては，その四角の透明エリアがボタンとして登録されているだけで，ボタンの絵柄としてそれらが登録されているわけではない。

ある1個のボタンを編集したいときはそのボタンをダブルクリックすれば画面は**図 8** のように変わる。これを見れば，ボタンの名前が“Go to Home”で，スタイルは“round rec”（四隅が丸い四角）であることがわかる。このボタンをクリックしたとき，このカードからホームカードに移れるようにリンクを張りたいとしよう。

それには，画面中の “Link To...” ボタンをクリックすると画面は**図 9** に変わる。どこへリンクしたいか聞いてくるので（このモードでは自動的に指マークへ変わっている），リンク先のホームカードに移動してから（**図 10**）“This Card”ボタンを押す。これにより，2枚のカードの間にリンクが張れた。この操作の結果，このリンクに対応するスクリプトが自動的に書かれている。ユーザー・レベルをスクリプティングに設定し，図8の“Script...”ボタンを押すと**図 11** のようなスクリプト編集画面になる。このスクリプトに，さらに編集を加えることもできる。

フィールドについて編集したいときは，ボタンのときと同じようにツールパレット最上段右のマークを選択する。

■ハイパートーク

ハイパートークとはハイパーカード・システムに内蔵のプログラミング言語で，オブジェクト指向である点が特徴である。ハイパークトークにおいては，ハイパーカード，ホームスタック，スタック，バックグラウンド，カード，ボタン，フィールドがオブジェクトである。これらのオブジェクトは階層構造をなしてい

◉ 図 6 は前ページ，図 12 は次ページ。

る(図 4)。

オブジェクト指向パラダイムでオブジェクトにアクションを起こさせる唯一の方法は，オブジェクトにメッセージを送ることである。メッセージとは「＊＊＊をしろ」という命令のことで，例えばユーザーはそれを言葉で言う代わりにマウスでクリックしたり，カーソルを移動させたりする。このようなキーやマウスの状況変化をマッキントッシュ・システムが察知し，それをオブジェクトにシステム・メッセージとして送る(**表 1**)。何も変化がない状況ではシステムは“idle”というシステム・メッセージを送り続けている。

「こういうメッセージが送られてきたら，こういうスクリプトを実行せよ」ということを記したものを，オブジェクトのメッセージハンドラとよぶ。あるオブジェクトがメッセージを受け取ったがそれに対応するメッセージハンドラをもっていない場合，そのメッセージは上位のオブジェクトに受け渡される。

図 12 にハイパートークで記述したスクリプトの例をあげておいた。ハイパートークはプログラム知識のないユーザーでもプログラムが書けるよう，さまざまな工夫がなされている。しかし，その目標が完全に達成されているとはまだ言い難い。以下に，かなり私的なハイパートークに対する感想を述べる。

- 他のプログラミング言語に比較して書きやすく，また読みやすい。しかし，プログラミング知識が全くないユーザーにとって，スクリプトを書くことはやはり困難である。
- 英語を母国語とするユーザーにとっては馴染みやすいが，そうでないとあまりメリットがない（例えば，英語の苦手な日本人ユーザー）。
- マッキントッシュのツールボックスの利用が簡単化された。半面，マッキントッシュへのシステム依存性は非常に高い。
- ハイパートークのグラマー(生成文法)は C や Pascal などの他言語に比較して整然としていない。明確な生成文法を覚えるというよりは,「こういうことがしたいときは，このようにスクリプトを書くのだ」というケース・バイ・

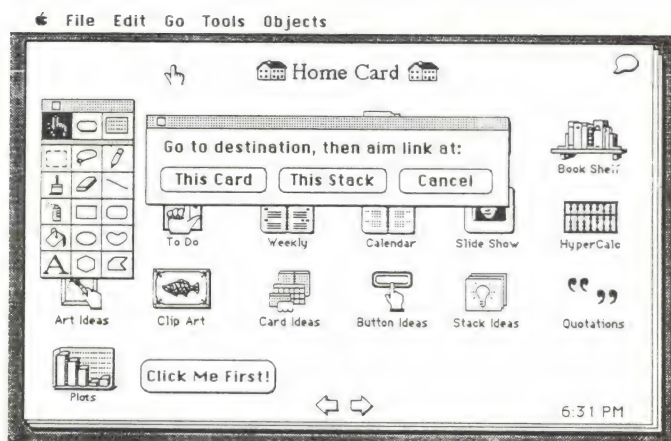



図 10

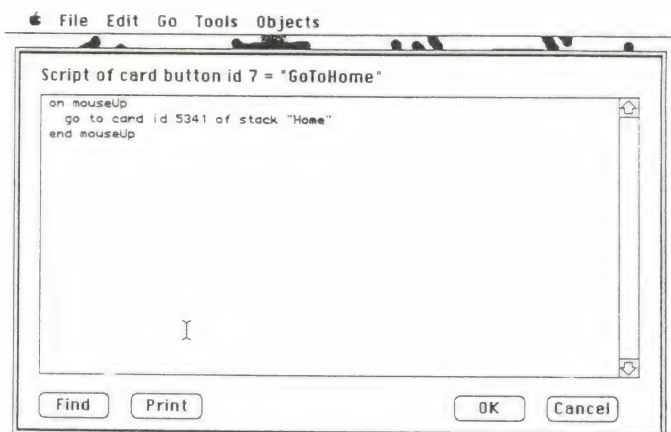



図 11

表 1 ハイパートークのシステム・メッセージ[6)]

マウスクリック	mouseDown, mouseStillDown, mouseUp
ボタン，フィールド内へのカーソルの出入り	mouseEnter, mouseWithin, mouseLeave
オブジェクト生成	newStack,newBkgnd,newCard,newField,newButton
オブジェクト消去	deleteStack, deleteBkgnd, deleteCard,…
別のスタックを開閉	openStack, closeStack
別のバックグラウンド／	openBkgnd, closeBkgnd
カードの表示開始と終了	openCard, closeCard
フィールド内へのテキスト挿入編集開始，終了(他のフィールド選択やエリア外でマウスクリック)	openField, closeField
キーの押し下げ	returnKey, enterKey, tabKey, arrowKey
指定メニューコマンドの起動(含むデスクアクセサリ)	doMenu　メニューコマンド
何も行われていない状態	idle
ハイパーカードの起動と終了	startUp(ハイパーカードからスタックの1番目のカードへ), quit
アプリケーションの起動と終了	suspend(ハイパーカードから他のアプリケーション起動), resume(起動アプリケーションからハイパーカードに戻った)

```
on mouseUp
  doMenu Opaque
  answer “いぢめる？” with “いぢめる” or “いぢめないよ”
  if it is “いぢめる” then
    set dragSpeed to 50
    put “10, 10” into point1
    put “100, 100” into point2
    repeat for 10
      beep 1
      choose lasso tool
      click at point1 with commandKey
      drag from point1 to point2
      drag from point2 to point1
    end repeat
    doMenu Transparent
  else
    doMenu Opaque
  end if
  choose browse tool
end mouseUp
```

図12 ハイパートークのスクリプト例

ケースのイディオムを暗記していく感が強い。他のプログラミング言語に習熟したプログラマにはしっくりこないかもしれない。

- 上記に関連するが，世の中に「ハイパートークのすぐに役立つスクリプト集」というような本が多数出版されているので，はじめからすべて自分で書こうとせずにそれらを手本として修正を加えるとよいであろう。

しかし，全体的にみると，ハイパートークのおかげでマッキントッシュ上のユーザー・インタフェース作成がずいぶん容易になり，大きな貢献をした。

★ハイパーメディアの問題点

ここでは，次世代ハイパーメディアに望むこととして，現状のハイパーメディアのかかえる問題点を考察してみる。まとめると以下のようになる。

(1) QUERY(問合せ)機能の付加
(2) データベースの統合
(3) プログラム知識がなくてもプログラミングを可能にするような支援ツール

最大の問題点は,「ハイパーメディアには全体の構造を見渡す機構がない」ことである。これは冒頭でも述べたように，拡張モデルとは並列的に統合型ユーザー・インタフェースとしてハイパーメディアが発展したことからくる当然の欠点である。これについて，ハイパーメディアとデータベースの比較を行ってみよう。

ハイパーメディアでは，ユーザーはリンクをたどることにより情報を選択する(navigational access)。つまり，そのカードまで行ってみないとその先が見えないので，目的の情報にたどり着くのが難しいことがある。また，一つ見つけた後も他にないか不安になる。前もって厳密な構造設計や制約の定義なしに気楽に使い始めることができる点がハイパーメディアの特徴であり，その結果，分類しにくいニュータイプの情報が出てきても扱うことができるというメリットがある。

データベースでは，厳密なスキーマ設計をはじめに行い，以後ユーザーは必要な情報を取り出したり，書き換えたり，変更したりするとき，データベース問合せ言語（query language）を用いてその内容を記述する。スキーマとはデータベースの情報構造を記述したものである。**図13**に，ある会社のスキーマと簡単な問合せ例を示す。データベースは構造化された情報なので，効率的検索が行えるというメリットがある。

以上のように，情報を厳格に構造化する/しないによって手軽さと効率の点で両者は相反する。ハイパーメディアでも適度に構造を取り入れ，データベースの問合せ機能に相当するものを組み込むことが必要である。単なるカードとリンクといったシンプルな構造から発展させて，多種類のリンクにより異なる意味を表したり，カードを階層化して複数カードを1枚のカードに集約したり，といった試みもなされている。他方，データベースの分野では，ポスト・リレーショナル・モデル研究の一環として，グラフ・データモデル[1]に加えて，あるいはそれを利用して，意味データモデルあるいはオブジェクト指向データベースの研究が盛んである。例えば，文献7)ではオフィスの秘書業務をケーススタディとして,「新しい種類の仕事が頻繁に発生する業種において，いかにその仕事管理をデータベース化するか，またスキーマの動的修正(普通,スキーマはあまり変更しないので静的といえる)をいかに行ったらよいか」を論じている。

ハイパーメディアとデータベースの融合には，G-BASE[2]のように，テキストだけでなく写真や絵などの映像を扱えるデータベースも必須となる。

第3番目の問題点「誰でもプログラムが作れるようにする支援ツールが欲しい」について考察してみる。一つのブレークスルーとして，テキストではなくアイコンなどでプログラムしていこうとする「ビジュアル・プログラミング」があるが，現状では以下のような問題点をかかえている。

■ビジュアル・プログラミングの問題点[8]

- スクリーンサイズが限られているので，大規模なプログラムや大量データが記述しにくい。これの対応策として，データの抽象化，階層化がある。
- プログラミングの対象が限定されてしまう。それは，あらかじめ扱えるオブジェクトやアクションを用意しておく必要があるからである。
- 構造化プログラミングがしにくい。Go To 文が多用される。
- フローチャートなどを可視化すると，リンクがこんがらがってスパゲッティのようになり，非常に見づらい。

●コメントが書けない。
● DBMS 機能が統合されていない。
●実行速度が遅い。

★おわりに

ハイパーカード以外のハイパーメディアについても少し述べておこう。

〈Xanadu〉[9]

Xanaduプロジェクトとは，世界中のすべての著作物をハイパーメディアの形式で格納し，オンラインで文献検索ができるようにしようとする大規模な構想である。実現させるためには，分散環境でのデータ格納・アクセスなどの問題の他，著作権の保護や課金問題などセキュリティに関する問題も解決しなくてはならない。

〈IBIS〉[10]

IBISではノードの型として「問題」「意見」「議論」の3種類があり，各ノードをリンクで接続していくことによりディスカッションが展開できるようになっている。IBISはテキストのみを扱うシステムである。

〈KMS〉[11]

大規模オンライン・マニュアルのハイパーテキスト・システムとしては，カーネギーメロン大学のKMSが有名である。高速にリンクをたどれる点が特徴である。

ハイパーメディアが期待される応用分野としては以下のようなものがある。

●ビジネス・プレゼンテーション
デモ・ビデオの制作に比べて制作が手軽である。
●複数ユーザーによるデータの蓄積→データベース作成
図書管理情報，レコード整理といった趣味のデータベースなど
●プロジェクト管理記録
会議議事録，報告書，回覧書類，新規参入者への説明にも使える。
●解説書，説明書（特にコンピュータ・マニュアル）
ユーザー・インタフェースなどは書籍で説明するよりハイパーメディアの方が適している。読み手が欲しい情報のみ選択できる点や対話型の面白さも魅力である。

1988年7月号のCACMのハイパーテキスト特集号では，ハイパーテキストの説明をハイパーテキストで頒布した[12]。筆者らも購入したが，論文の著者の顔写真などから「こういう人が書いていたのか」とわかって，論文に対する親しみがわいてきた。筆者らもボストンでの1989年9月のヒューマン・インタフェース国際会議では，論文内容説明をハイパーカード化し，興味ある研究者には「マッキントッシュで見て下さい」と言って渡す用意をしているところである。ハイパーメディアは手軽だが効果的な説明用メディア，ならびに多様な情報を統合化するユーザー・インタフェースとして，今後も広く普及・発展していくと考えられる。

以上のハイパートークに関する感想には，マッキントッシュ・ユーザー仲間であるチュールスト・マーティン氏（國井研究室の大学院生）の意見も含まれていることを付記し，謝辞に代える。

社員のレコード・タイプ：
社員（社員No.，氏名，住所，部署）
レコードタイプ　属性1　属性2　属性3　属性4

100	山田太郎	目白	営業
101	山本次郎	本郷	総務
102	児島弘子	青山	人事

DATABASE SCHEMA SAMPLE
部署
管理
所属
社員 マネージャ
□：record type
――：link type

```
「上司と部下の関係にある全ての社員ペアの氏名をプリントせよ」
    database：ABC会社；
    alias：社員　マネージャ；
    target 上司部下　マネージャ，部署，社員；
    find 上司部下＝
        マネージャ．．／管理／．．部署．．／所属／．．社員
            ［マネージャ．社員No！＝社員．社員No］；
    for 上司部下 print マネージャ．氏名，社員．氏名；
```

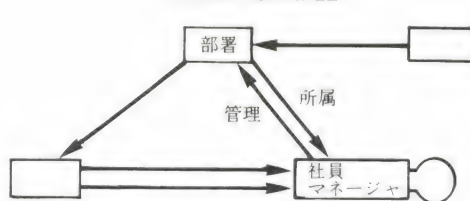



図13　リコーのG-BASE[2]を用いた ある会社のスキーマ例と問合せの例

参考文献

1) Kunii,H.S.（國井秀子）："Graph Data Language：A High Level Access－Path Oriented Language",Ph.D.Dissertation to the Department of Computer Science,the University of Texas at Austin,1983
2) G-BASEマニュアル,Vol. 1-4, リコー・ソフトウェア研究所, 1988
3) Knuth,D.E.："The Art of Computer Programming,Vol.1：Fundamental Algorithms",Addison-Wesley Publishing Company, Reading,Massachusetts,Amsterdam,1973
4) 金崎克己，國井利泰：「ハイパーメディアの世界への招待」，プロンプト，4，12，日刊工業新聞社，pp.7-14，1988
5) Markman,M.J.："Epilogue：HyperCard", in Interactive Multimedia(Visions of Multimedia for Developers,Educators& information Providers), Ambron,S.,Hooper,K.(ed.),Microsoft Press,Washington,pp.331-339,1988
6) Goodman,D.："The Complete HyperCard handbook (2nd Edition)",Bantan Books,Toronto, New York,1988
7) Kanasaki,K.(金崎克己),Kunii,T.L.(國井利泰)："Case-Based Evolutionary World Model for Electronic Secretaries",リコー・ソフトウェア研究所テクニカル・レポート, 1988
8) Myers,B.A.："Creating User Interface by Demonstration", Academic Press,Inc.,San Diego, CA,1988
9) Nelson,T.H.："Literary Machines",自費出版,1987
10) Begeman,M.L.,Conklin,J.："The Right Tool for the Job",BYTE,pp.255-266,1988
11) Akscyn,D.L.,McCracken,D.L.,Yoder,E.A.："KMS：A Distributed Hypermedia System for Managing Knowledge in Organizations",CACM,pp.820-835,1988
12) "Communications of ACM Special Issue on Hypertext on floppy disks",ACM Press, 1988.7

連載

Cプログラミングを用いたレンダリング・ソフトの実践シリーズ〔最終回〕

今回はこのシリーズの最終回として，
- 簡単なプリミティブ作成プログラムの紹介
- ツリー構造についての補足説明
- debut を使ったテスト・レンダリングと作品例，およびその制作の実際
- debut プログラムの発展

について順次説明していきたいと思う。

出渕　亮一朗*

プリミティブ作成ツール makeprim

表1 のソース・プログラムは，debut で使用できるオブジェクト・ファイル.obj を生成するためのツールである。.obj はそのデータ形式さえ正しければ，自分で図面を引くなりして手で打ち込んで作成してもよいし，このようにプログラムによって作成してもよい。また，このプログラムにかぎらず，自分でいろいろ研究して新しいプリミティブを付け加えることも可能である。さらに高度なワザとして，自然法則に従ったモデリングをオブジェクト生成プログラムにより行うこともできる。

makeprim はそれらの参考となるべきごく単純なプログラムである。これもC言語で記述されているが，使用方法は起動した後モニターに出てくるメッセージに順次従って，自分の好みのデータを打ち込んでいくだけでよい。＜＞の中にはデフォルト値が示されているので，もしもその値でよいと思った場合は，そのままリターンキーを押せばその値が入力される。**表2** にモニター出力例をあげておく（この例では，三角分割された各辺100のデフォルトの Cube が default.obj の名前で作成されている）。

makeprim では，Mesh，N-Polygon，Cube, Rotation shape, Sphere, Cylinder (Corn)，Coil，Spiral の8つのプリミティブが用意されている（**図1**)。Mesh は単純な長方形の面を作成する。N-Polygon は N 多角形を作成するが，これは後述する Cylinder や Rotation shape の蓋に主に使用できる。なぜなら dubut の .obj では，1つのオブジェクトはスムーズ・シェーディングがすべてかかるか，全くかからないかのどちらかしか選択できないので，円柱のような部分的にスムーズ・シェーディングの有る無しが混在するオブジェクトは，.obj を分割して組み合わせなければならないからである。また，Mesh，N-Polygon で作成した.obj は面であるので，法線がどちらを向いているかを常に注意しなければならない(以前，述べたように，法線が逆向きの場合は表示されない)。

Cube は直方体を生成する。Rotation shape は回転体を生成するが，このタイプの使用法は他と違って少し複雑である。まず，回転の基線となる曲線のためのデータを作る必要がある。これにはまた .obj を使う。ただし，今回新たに LN という.obj の中のラベルを設定する。これは SF と同様であるが，SF のように閉じた多角形ではなくラインの形であることを示すものである。これを使って xy 平面上に y の正の方向から負の方向に向かうラインを設定する（逆方向に作れ

◉表1は159～164ページ参照。

表2 makeprim モニター出力例

```
#
# makeprim.c input date sample
# (This is monitor outputs.)
# making default cube
#
SELECT OBJECT TYPE
Rectangle Mesh = 1
N-Polygon     = 2
Cube          = 3
Rotation Shape = 4
Sphere        = 5
Cylinder(Corn) = 6
Coil          = 7
Spiral        = 8
Object Type=<3>
MAKE CUBE
Input Size X=<100>
Input Size Y=<100>
Input Size Z=<100>
Which? Triangle or Rectangle Patch(t/r)=<t>
Atach Material?(y/n)=<y>
Input .mat File Name=<mat>
Input Material Name=<col>
Use Mapping?(y/n)=<n>
INPUT SAVE OBJECT FILE NAME
Save File Name=<default>
```

*でぶち　りょういちろう　〒154　東京都世田谷区三宿1-15-11　コート世田谷101

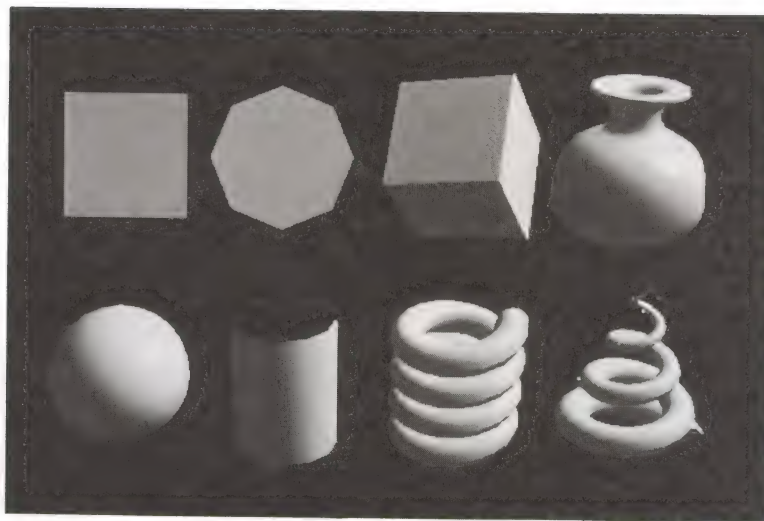



図1 makeprim によるプリミティブ

ば生成される回転体の法線が内側を向いてしまうので注意！）。この.obj をツール内で読み込んで回転させればよいのである。また，読み込んだ曲線をスプラインで補間してから回転体を作るようにプログラムを発展させれば，さらに良くなると思う。makeprim では，y 軸まわりの回転体のみしか作ることができない。しかし，もし他の軸方向に向けたい場合は，以前述べたように debut の ver.0 を使用すればよい。ver.0 はワイヤー作成ツールであるが，.obj を生成する一種のモデラーとしても使用できる。.str で例えば z 軸で90°回す指定を行い，これを用いて makeprim で作成した回転体を debut ver.0 に入力すれば z 軸回転が行われた.obj が出力されるわけである（また，もちろん debut ver.0 はいくつかの.obj を好きなように組み合わせて1つの.obj に出力することができるので，これと makeprim を併用すればかなりいろいろなモデリングが行えると思う）。

Sphere は球を，Cylinder は円柱または円錐を作成する。ここで scale を聞いてくるのは，x，y，z 方向の拡大・縮小率のことである。Coil は写真のようにコイルを作るものであるが，U 回転を 360°，高さを 0 とすればドーナツ形（トーラス）にすることもできる。Spiral は Coil の変形で螺旋形を作成する。実は，このバリエーションはいろいろ考えることができ，例えば，V 方向の半径を周期的に変えれば巻貝のような形を作ることができる。

makeprim ではまた，マッピング・バーテックス（VM）を取り付けることを選択できる。どのタイプのオブジェクトにも単純な VM が取り付けられるようになっている。マテリアル・ファイル.mat へのインデックスを入力し，ファイル名を打ち込めば完成である。

ツリー構造データの補足説明

オブジェクトのツリー構造は，木構造，階層構造，クラスターなどさまざまな呼び名があるが，筆者はコンピュータ・グラフィック・アルゴリズムの技術の中でも最も重要なものの一つで，またわかりにくいものであると思う。しかし不思議なことに現在までに出版されたいくつかのコンピュータ・グラフィックス（CG）に関する本や雑誌の中で，このことについて詳しく説明されているものはごくわずかである。そのためかアマチュアの作品やプログラムの中には，一見してツリー構造を使用していないということがわかるものがある。ツリー構造は CG の画像を制作するうえで重要であるし，非常に便利なものであるので，連載1，2（88年6月号，9月号）に引き続き，もう一度復習の意味を込め，実際に絵を作るデザイナーの側に立って説明してみたいと思う。

この世界のすべてのものは意味のうえで記号化することができるが，記号にはより大きな重要なものと，一つの記号の中に集まっているさらに細かな記号というように，いくつもの階層レベルで考えることができる。CG で現実物のモデリングを行う場合も，脳はそのように認知するよう働いているし，実際，人工物や自然物もそのような構造となっている。例えば，オープンカーに乗った人物がビルの並び立つ市街を走っているシーンを作るとしよう。まず，道路・ビル・車・人物といったオブジェクトを別々に作り，人を車の上に乗せ，その車を道路とビルを組み合わせた市街の中の適当な位

図3 tree 1～tree 6 のためのツリー構造図

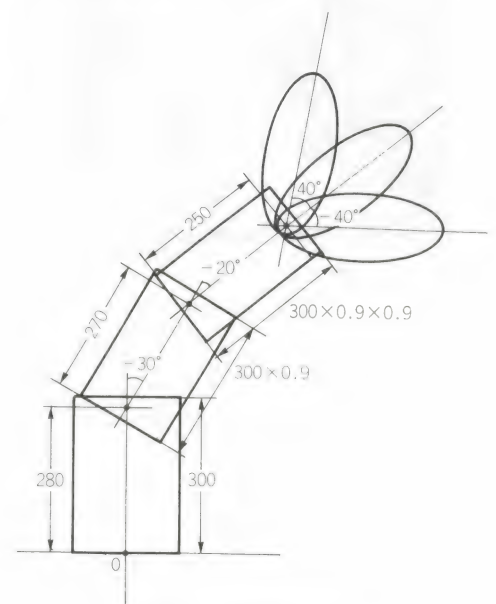



図4 tree 構造説明 tree

表 3　tree 構造 1

```
# tree1.str
- - ROOT t.cam K  0 0 700 0 0 0 0
            * > ground ground.obj O xyz 0 0 0 -100 -200 -100 0.3
            * > lit1 spot.lit L xyz 0 0 0  1000 1000 500  1
            ground > branch1 coil.obj O /D
            * > branch2 coil.obj O /D
            * > branch3 coil.obj O /D
                        * > leaf1 leaf.obj O /D
                        * | leaf2 leaf.obj O /D
                        * | leaf3 leaf.obj O /D
```

表 4　tree 構造 2

```
# tree2.str
- - ROOT t.cam K 0 0 700 0 0 0 0
            * > ground ground.obj O xyz 0 0 0 -100 -200 -100 0.3
            * > lit1 spot.lit L xyz 0 0 0  1000 1000 500  1
            ground > branch1 coil.obj O xyz 0 0 0 0 0 0 1
                        * > branch2 coil.obj O xyz 0 0 -30 0 0 0 1
                                    * > branch3 coil.obj O xyz 0 0 0 0 0 0 1
                                                * > leaf1 leaf.obj O xyz 0 0 0 0 0 0 1
                                                * | leaf2 leaf.obj O xyz 0 0 0 0 0 0 1
                                                * | leaf3 leaf.obj O xyz 0 0 0 0 0 0 1
```

表 5　tree 構造 3

```
# tree3.str
- - ROOT t.cam K 0 0 700 0 0 0 0
            * > ground ground.obj O xyz 0 0 0 -100 -200 -100 0.3
            * > lit1 spot.lit L xyz 0 0 0  1000 1000 500  1
            ground > branch1 coil.obj O xyz 0 0 0 0 0 0 1
                        * > branch2 coil.obj O xyz 0 0 -30 0 280 0 0.9
                                    * > branch3 coil.obj O xyz 0 0 0 0 0 0 1
                                                * > leaf1 leaf.obj O xyz 0 0 0 0 0 0 1
                                                * | leaf2 leaf.obj O xyz 0 0 0 0 0 0 1
                                                * | leaf3 leaf.obj O xyz 0 0 0 0 0 0 1
```

図 2　tree 構造説明 tree（①～⑥）

① tree 1

② tree 2

③ tree 3

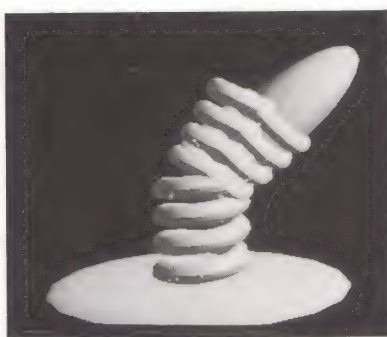
表 6　tree 構造 4

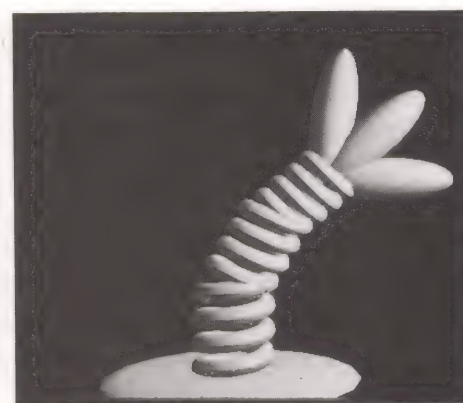
```
# tree4.str
- - ROOT t.cam K 0 0 700 0 0 0 0
            * > ground ground.obj O xyz 0 0 0 -100 -200 -100 0.3
            * > lit1 spot.lit L xyz 0 0 0  1000 1000 500  1
            ground > branch1 coil.obj O xyz 0 0 0 0 0 0 1
                        * > branch2 coil.obj O xyz 0 0 -30 0 280 0 0.9
                                    * > branch3 coil.obj O xyz 0 0 -20 0 0 0 1
                                                * > leaf1 leaf.obj O xyz 0 0 0 0 0 0 1
                                                * | leaf2 leaf.obj O xyz 0 0 0 0 0 0 1
                                                * | leaf3 leaf.obj O xyz 0 0 0 0 0 0 1
```

表 7　tree 構造 5

```
# tree5.str
- - ROOT t.cam K 0 0 700 0 0 0 0
            * > ground ground.obj O xyz 0 0 0 -100 -200 -100 0.3
            * > lit1 spot.lit L xyz 0 0 0  1000 1000 500  1
            ground > branch1 coil.obj O xyz 0 0 0 0 0 0 1
                        * > branch2 coil.obj O xyz 0 0 -30 0 280 0 0.9
                                    * > branch3 coil.obj O xyz 0 0 -20 0 270 0 0.9
                                                * > leaf1 leaf.obj O xyz 0 0 0 0 0 0 1
                                                * | leaf2 leaf.obj O xyz 0 0 0 0 0 0 1
                                                * | leaf3 leaf.obj O xyz 0 0 0 0 0 0 1
```

表 8　tree 構造 6

```
# tree6.str
- - ROOT t.cam K 0 0 700 0 0 0 0
            * > ground ground.obj O xyz 0 0 0 -100 -200 -100 0.3
            * > lit1 spot.lit L xyz 0 0 0  1000 1000 500  1
            ground > branch1 coil.obj O xyz 0 0 0 0 0 0 1
                        * > branch2 coil.obj O xyz 0 0 -30 0 280 0 0.9
                                    * > branch3 coil.obj O xyz 0 0 -20 0 270 0 0.9
                                                * > leaf1 leaf.obj O xyz 0 0 0 0 250 0 0.85
                                                * | leaf2 leaf.obj O xyz 0 0 -40 0 250 0 0.85
                                                * | leaf3 leaf.obj O xyz 0 0 40 0 250 0 0.85
```

④ tree 4

⑤ tree 5

⑥ tree 6

置に配置させるであろう。このとき人物は頭・胴体・腕など，またそのレベルでさまざまなパーツにより組み合わされているはずである。モデルを考えるうえでは，一度人物のモデルを組み立ててしまえば，腕が肩からどんな角度で取り付けられているかなどという細かいことは忘れて「人物」として車の上に乗せたいものである。そして人と車を一体として道路上のしかるべき位置に配置させたいものである。もしもツリー構造データがないとすれば，人物の手の先の指も巨大な

◉図 3，4 は 155 ページ。

ビルも全く同じレベルで空間の中のどこにどんな角度で配置されるかを考えるという，信じられない作業を行わなければならない。そのとき車は適当に回転平行移動を受け，人物もまた車に対して相対的に回転平行移動しているというようなことをすべて考えなければならないのである。

さて，それでは実際のツリー構造データを，debut の.str をもとに具体的に説明してみよう（**図 2〜4，表 3〜8**）。プログラムの中では 1 つの node においては，スケーリング→回転→平行移動のプライオリティで変換が行われることを覚えておいてほしい。また回転についても，x, y, z 軸のまわりの回転をどの順で組み合わせるかというプライオリティも時には重要である。図 2-①は ground の上に branch 3 本，leaf 3 個が同一の位置にあるはじめの状態である。図 2-②で branch 2 を z 軸に 30°回してみた。図 2-③ではこれを y 軸方向に 280 平行移動した。もしも，branch 1 と branch 2 が同じレベルにあれば，branch 2 は x と y 方向のしかるべき位置に移動させるという面倒なことをしなければならないが，branch 2 は branch 1 の下位レベルにあるので，単純に branch 1 の長さの分だけ y 方向に移動するだけでよいのである（なお，これらの branch.obj, leaf.obj は，原点の位置を y 軸の下の方という都合の良い位置に設定してあるので，普通は y 方向に平行移動してから z 軸に対して回転させるという順序をとった方がやりやすい）。図 2-④，⑤で同様に branch 3 を branch 2 の先に移動して取り付けた。図 2-⑥では，branch 3 の下にある同じレベルの 3 つの leaf.obj を branch 3 の先に z 軸に対して，それぞれ 40°，0°，−40° 回転させ広げて取り付けている。

また，ツリー構造データはアニメーションを行う場合，重要である。例えば，遊園地のティーカップの運動などはツリー構造なしでは絶対に作れない。ティーカップはそれぞれのティーカップが回転しつついくつかのティーカップを乗せたまとまりがさらに回転しているといった，まさにツリー構造そのものの運動を行っている。

今回はポリゴンをもとにしたデプス・バッファのオブジェクトに対してのツリー構造を考えたが，もちろん，レイ・トレーシングのプリミティブについても同様にツリー構造を用いて組み立てることができる。平面や直方体はよいとして，問題と思われるのは 2 次曲面の処理であると思う。この場合は，次のように考えれば 4×4 マトリクスで変換した後の 2 次曲面式が求められる[1]。2 次曲面

$$Ax^2+By^2+Cz^2=1 \qquad (1)$$

を 4×4 変換行列 M で変換したときの曲面式を

$$f(x,\ y,\ z)=ax^2+by^2+cz^2+dxy+eyz+fzx+gx+hy+iz+j=0 \qquad (2)$$

であるとする。M の逆行列を

$$M^{-1}=R=\begin{bmatrix} R_{11} & R_{12} & R_{13} & R_{14} \\ R_{21} & R_{22} & R_{23} & R_{24} \\ R_{31} & R_{32} & R_{33} & R_{34} \\ R_{41} & R_{42} & R_{43} & R_{44} \end{bmatrix}$$

(1) 上の点を

$$P=\begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

(2) 上の点を

$$P'=\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix}$$

とすると，

$$P'=PM,\quad P=P'M^{-1}=P'R \qquad (3)$$

(3)より

表 10 "Coils"のための .str データ

```
#
# coil.str -- .str for Coils & Light Test
#
- - ROOT t.cam K 0 0 687 0 0 0 0
 ROOT > stage default D yxz 33 26.9 0 0 50 -950 1
  * > lights default D xyz 0 0 0 0 -400 0 1
   * > lit1 default D xyz 0 0 0 0 514 0 1
    * > lt1 light1M.obj O xyz 180 0 0 0 0 0 0.100045
     * > light1 spot1.lit L /D
   lit1 | lit2 default D xyz -40 0 0 0 0 0 1
    * > lt2 light2M.obj O xyz -90 0 0 0 0 405 0.100045
     * > light2 spot2.lit L /D
   lit1 | lit3 default D xyz -40 60 0 0 0 0 1
    * > lt3 light3M.obj O xyz -90 0 0 0 0 405 0.1
     * > light3 spot3.lit L /D
   lit1 | lit4 default D xyz -40 120 0 0 0 0 1
    * > lt4 light4M.obj O xyz -90 0 0 0 0 405 0.1
     * > light4 spot4.lit L /D
   lit1 | lit5 default D xyz -40 180 0 0 0 0 1
    * > lt5 light5M.obj O xyz -90 0 0 0 0 405 0.1
     * > light5 spot5.lit L /D
   lit1 | lit6 default D xyz -40 240 0 0 0 0 1
    * > lt6 light6M.obj O xyz -90 0 0 0 0 405 0.1
     * > light6 spot6.lit L /D
   lit1 | lit7 default D xyz -40 300 0 0 0 0 1
    * > lt7 light7M.obj O xyz -90 0 0 0 0 405 0.1
     * > light7 spot7.lit L /D
stage > wall1 wallM.obj O xyz 0 0 0 0 0 -400 1
 * | wall2 wallM.obj O xyz 0 -90 0 400 0 0 1
 * | floor wallM.obj O xyz -90 0 0 0 -400 0 1
 * | coil1 coilM.obj O xyz 0 0 0 -146 -400 -146 0.701131
 * | coil2 coil2M.obj O xyz 0 0 0 -146 -400 146 0.7
 * | coil3 coil3M.obj O xyz 0 0 0 146 -400 -146 0.7
 * | coil4 coil4M.obj O xyz 0 0 0 146 -400 146 0.7
```

表 12 .cam データ

```
#
# t.cam
#
FL .4
SZ 636 482
CY      5000
```

◀**図 7 "Cat & Dog"のワイヤー・モデル**

表 13 .lit データ 1

```
#
# spot.lit
#
LT d
LI 3 3 3
DP 0.1
```

表 14 .lit データ 2

```
#
# spot2.lit
#
LT d
# LI 1.5 1.5 1.5
LI 3 0 0
DP 0.3
```

＊その他の色のスポットライトも同様.

◉図 5, 6 は COLOR IMAGES,図 7, 表 10, 12～14 は前ページ参照。

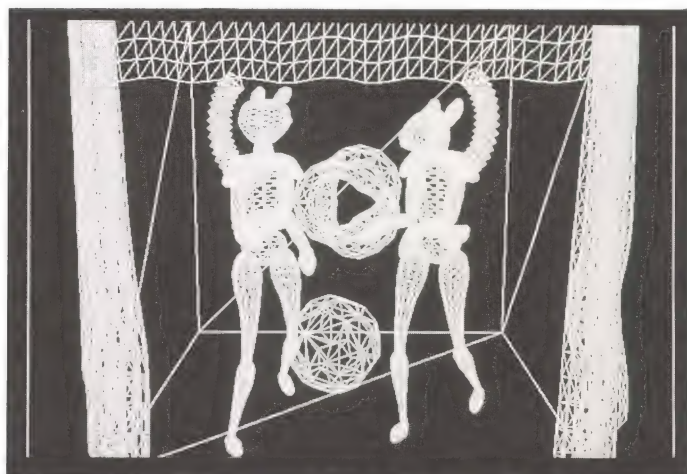



表 9 “Cat & Dog”のための .str データ

```
#
# ****** stagel.str --- .str for Cat & Dog ********************
#
- - ROOT t.cam K 0 0 687 0 0 0 0
.ROOT > stage default D xyz 9 0 0 0 50 -950 1
  * > lit1 spot.lit L xyz 0 0 0 1000 1000 500 1
   lit1 | lito2 lightM.obj O xyz 0 0 -114.5 55 180 350 0.100045
     * > lit2 spot2.lit L /D
   lit1 | lito3 lightM.obj O xyz 0 0 33.2 -44 101 300 0.100045
     * > lit3 spot3.lit L /D
   lit1 | lito4 lightM.obj O xyz 0 0 180 -51 -126 70 0.100045
     * > lit4 spot4.lit L /D
   lit1 | lito5 lightM.obj O xyz -21.7 0 -20.6 16 -196 400 0.100045
     * > lit5 spot5.lit L /D
   lit1 | lito6 lightM.obj O xyz -90 0 0 -57 77 -286 0.100045
     * > lit6 spot6.lit L /D
   lit1 | lito7 lightM.obj O xyz -90 0 0 247 -259 -300 0.100045
     * > lit7 spot7.lit L /D
   lit1 | lito8 lightM.obj O xyz 0 0 -111.8 -513 100 600 0.100045
     * > lit8 spot2.lit L /D
   lit1 | lito9 lightM.obj O xyz 0 0 114.3 600 50 600 0.100045
     * > lit9 spot3.lit L /D
   stage > wall1 wallM.obj O xyz 0 0 0 0 0 -400 1
    * | wall2 wallM.obj O xyz 0 -90 0 400 0 0 1
    * | wall3 wallM.obj O xyz 0 90 0 -400 0 0 1
    * | wall4 wallM.obj O xyz 90 0 0 0 400 0 1
    * | wall5 wallM.obj O xyz -90 0 0 0 -400 0 1
   stage > ball mariballM.obj O xyz 0 0 0 -36 -305 0 1.17382
    * | knot mariknotM.obj O xyz 0 0 0 0 0 0 1
    * | curt1 curt1M.obj O xyz 14.1 0 0 0 383 400 1
    * | curt2 curt2M.obj O xyz 0 66 0 394 0 400 1.09496
    * | curt3 curt2M.obj O xyz 0 -69.8 0 -389 0 400 1.10662
   stage > dog default D xyz 0 134.9 0 157 33 200 0.250148
     * > body mari10M.obj O xyz 0 0 0 0 0 0 1
     * > arms default D xyz 0 0 0 0 221 0 1
     * > sho1 mari4M.obj O xyz 0 0 -39.9 195 0 0 0.533334
     * > arm1 mari5M.obj O xyz 0 0 45.8 0 0 0 2.28245
     * > arm1b mari5M.obj O xyz 90.4 0 0 0 -336 0 0.860993
     * > hand1 mari3M.obj O xyz -32.1 0 0 0 -312 0 0.627479
      arms > sho2 mari4M.obj O xyz 0 -7.4 35.3 -195 0 0 0.533825
       * > arm2 mari5M.obj O xyz 26.5 0 -50.7 0 0 0 2.28245
        * > arm2b mari5M.obj O xyz 52.5 0 0 0 -336 0 0.860993
         * > hand2 mari3M.obj O xyz -44 0 0 0 -312 0 0.627768
     body > hip mari6M.obj O xyz 0 12.7 0 0 -344 0 1.40088
       * > foots default D xyz 0 0 0 0 -64 0 1
        * > momo1 mari7M.obj O xyz 29.9 0 0 110 0 0 1
         * > nie1 mari2M.obj O xyz 0 0 0 0 -550 95 3.99578
          * > leg1 mari8M.obj O xyz -39.7 0 0 0 0 0 0.202394
           * > foot1 mari3M.obj O xyz 83.2 0 0 0 -477 0 0.457719
      foots > momo2 mari7M.obj O xyz -2.2 0 0 -100 0 0 1
         * > nie2 mari2M.obj O xyz 0 0 0 0 -550 95 3.99578
          * > leg2 mari8M.obj O xyz -39.7 0 0 0 0 0 0.202394
           * > foot2 mari3M.obj O xyz 83.2 0 0 0 -477 0 0.457719
     body > neck mari3M.obj O xyz 0 0 0 0 538 0 0.809468
       * > head mari1M.obj O xyz 67 -5.2 0 0 0 0 1
        * > ear default D xyz 141.6 0 0 0 0 0 0.673286
         * > ear1 mari3M.obj O xyz 0 0 10.6 164 0 0 1
       ear > ear2 mari3M.obj O xyz 0 0 -10 -164 0 0 1
      head > eye default D xyz 0 0 0 0 -140 -46 3.99947
         * > eye1 mari2M.obj O xyz 0 0 0 38 0 0 1
         eye > eye2 mari2M.obj O xyz 0 0 0 -38 0 0 1
      head > nose mari2M.obj O xyz 0 0 0 0 -440 0 3.99947
       body > tail mari3M.obj O xyz -106.1 0 0 0 -340 107 0.778394
       body > tube maritubeM.obj O xyz 0 -108.2 90 0 615 -47 2.99779
   stage > cat default D xyz 0 -138.2 0 -160 24 180 0.250148
    * > cbody mari10bM.obj O xyz 0 0 0 0 0 0 1
     * > carms default D xyz 0 0 0 0 221 0 1
      * > csho1 mari4bM.obj O xyz 0 0 -39.9 195 0 0 0.533334
       * > carm1 mari5bM.obj O xyz 20.8 0 60.4 0 0 0 2.28245
        * > carm1b mari5bM.obj O xyz 90.4 0 0 0 -336 0 0.860993
         * > chand1 mari3bM.obj O xyz -32.1 0 0 0 -312 0 0.627479
     carms > csho2 mari4bM.obj O xyz 0 -7.4 35.3 -195 0 0 0.533825
      * > carm2 mari5bM.obj O xyz -27.4 0 -50.7 0 0 0 2.28245
       * > carm2b mari5bM.obj O xyz 52.5 0 0 0 -336 0 0.860993
        * > chand2 mari3bM.obj O xyz -44 0 0 0 -312 0 0.627768
    cbody > chip mari6bM.obj O xyz 0 -10 0 0 -344 0 1.40088
     * > cfoots default D xyz 0 0 0 0 -64 0 1
      * > cmomo1 mari7bM.obj O xyz 29.8 0 10 110 0 0 1
       * > cnie1 mari2bM.obj O xyz 0 0 0 0 -550 95 3.99578
        * > cleg1 mari8bM.obj O xyz -39.7 0 0 0 0 0 0.202394
         * > cfoot1 mari3bM.obj O xyz 83.2 0 0 0 -477 0 0.457719
     cfoots > cmomo2 mari7bM.obj O xyz -2.2 0 -12.5 -100 0 0 1
         * > cnie2 mari2bM.obj O xyz 0 0 0 0 -550 95 3.99578
          * > cleg2 mari8bM.obj O xyz -39.7 0 0 0 0 0 0.202394
           * > cfoot2 mari3bM.obj O xyz 83.2 0 0 0 -477 0 0.457719
    cbody > cneck mari3bM.obj O xyz 0 0 0 0 447 0 0.746792
     * > chead mari9bM.obj O xyz -10.7 -5.1 0 0 0 0 1
      * > cear default D xyz 141.6 0 0 0 344 0 0.673596
       * > cear1 mari3bM.obj O xyz 0 0 10.6 164 0 0 1
      cear > cear2 mari3bM.obj O xyz 0 0 -10 -164 0 0 1
      chead > ceye default D xyz 0 0 0 0 215 -158 3.99763
       * > ceye1 mari2bM.obj O xyz 0 0 0 38 0 0 1
        ceye > ceye2 mari2bM.obj O xyz 0 0 0 -38 0 0 1
      chead > cnose mari2bM.obj O xyz 0 0 0 0 104 -155 3.99763
    cbody > ctail mari3bM.obj O xyz -106.1 0 0 0 -340 107 0.778394
    cbody > ctube maritubeM.obj O xyz 0 -108.2 90 0 615 -47 2.99779
```

表 11 .mat データ

```
#
# mat.mat
#
NEW mari1
AMB 0 0 0
DIF 0 0 0
SPC 2 2 2
EXP 70
MDI wood.rgb 1 1 .5
MAM wood.rgb 1 1 .1
SMS on
LTM 2
#
NEW mari2
AMB 0 0 0
DIF 0 0 0
SPC 2 2 2
EXP 70
MDI tile.rgb 1 1 .5
MAM tile.rgb 1 1 .1
SMS on
LTM 3
#
NEW wall
AMB 0 0 0
DIF 0 0 0
SPC 2 2 2
EXP 50
MDI check.rgb 1 1 .3
MAM check.rgb 1 1 .1
SMS on
LTM 3
#
NEW curt
AMB 0 0 0
DIF 0 0 0
SPC 2 2 2
EXP 70
MDI flow.rgb 1 1 .6
MAM flow.rgb 1 1 .1
SMS on
LTM 3
#
NEW red
AMB .1 .1 .1
DIF 1 0 0
SPC 2 2 2
EXP 100
SMS on
LTM 3
#
NEW blue
AMB .1 .1 .1
DIF 0 0  1
SPC 2 2 2
EXP 100
SMS on
LTM 3
#
NEW white
AMB .1 .1 .1
DIF .5 .5 .5
SPC 2 2 2
EXP 70
SMS on
LTM 3
#
NEW redlight
AMB .8 0 0
DIF .2 0 0
SPC .9 .9 .9
EXP 100
SMS on
LTM 3
#
NEW bluelight
AMB 0 0 .8
DIF 0 0 .2
SPC .9 .9 .9
EXP 100
SMS on
LTM 3
#
NEW greenlight
AMB 0 .8 0
DIF 0 .2 0
SPC .9 .9 .9
EXP 100
SMS on
LTM 3
#
NEW yellowlight
AMB .8 .8 0
DIF .2 .2 0
SPC .9 .9 .9
EXP 100
SMS on
LTM 3
#
NEW cyanlight
AMB 0 .8 .8
DIF 0 .2 .2
SPC .9 .9 .9
EXP 100
SMS on
LTM 3
#
NEW magentalight
AMB .8 0 .8
DIF .2 0 .2
SPC .9 .9 .9
EXP 100
SMS on
LTM 3
#
NEW ball
AMB .1 .1 .1
DIF  .5 .5 .5
SPC 2 2 2
SMS off
EXP 50
LTM 3
#
NEW light
AMB .8 .8 .8
DIF .2 .2 .2
SPC .9 .9 .9
EXP 100
SMS on
LTM 3
#
NEW coil
AMB .1 .1 .1
DIF  .5 .5 .5
SPC 2 2 2
SMS off
EXP 70
LTM 3
#
NEW coil2
AMB .1 .1 .1
DIF  .5 .5 .5
SPC 2 2 2
SMS on
EXP 70
LTM 3
#
NEW coil3
AMB 0 0 0
DIF 0 0 0
SPC 2 2 2
MAM flow.rgb 8 1 .1
MDI flow.rgb 8 1 .5
SMS off
EXP 70
LTM 3
#
NEW coil4
AMB 0 0 0
DIF 0 0 0
SPC 2 2 2
MAM flow.rgb 8 1 .1
MDI ilow.rgb 8 1 .5
SMS on
EXP 70
LTM 3
```

$$\begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} R$$

と P が P'を用いて表せるので，これを(1)式に代入して整理し，(2)式と係数を比較すると次式が求まる。

$$\begin{aligned}
a &= AR_{11}{}^2 + BR_{12}{}^2 + CR_{13}{}^2 \\
b &= AR_{21}{}^2 + BR_{22}{}^2 + CR_{23}{}^2 \\
c &= AR_{31}{}^2 + BR_{32}{}^2 + CR_{33}{}^2 \\
d &= AR_{11}R_{21} + BR_{12}R_{22} + CR_{13}R_{23} \\
e &= AR_{21}R_{31} + BR_{22}R_{32} + CR_{23}R_{33} \\
f &= AR_{11}R_{31} + BR_{12}R_{32} + CR_{13}R_{33} \\
g &= AR_{11}R_{41} + BR_{12}R_{42} + CR_{13}R_{43} \\
h &= AR_{21}R_{41} + BR_{22}R_{42} + CR_{23}R_{43} \\
i &= AR_{31}R_{41} + BR_{32}R_{42} + CR_{33}R_{43} \\
j &= AR_{41}{}^2 + BR_{42}{}^2 + CR_{43}{}^2 - 1
\end{aligned} \tag{4}$$

このマトリクス M で変化させた場合の2次曲面式(4)を用いて，同様に光線との交点や内外判別を行えばよいのである。例えば，法線ベクトルは関数を偏微分すればよいから

$$\begin{aligned}
\partial f / \partial x &= ax + dy + fz + g \\
\partial f / \partial y &= by + ez + dx + h \\
\partial f / \partial z &= cz + fx + ey + i
\end{aligned} \tag{5}$$

が求まる。もしも，お手もとのソフトウエアにツリー構造のない方がいたら，ぜひ取り付けてみることをお勧めする。

debut を用いた作品制作の実際

図 5，6 は，debut ver.2 を用いた作品“Cat & Dog (Marionette Show)”と“Coils”である。**図 7** は，“Cat & Dog”のためのワイヤー・モデルを debut ver.0 で計算したものである。また，“Cat & Dog” “Coils”のための.str データ，.mat データ，.cam データ，.lit データをそれぞれ掲載する（**表 9～14**）(.obj のデータは長くなるので省略する)。

“Coils”はさまざまなマテリアルの例とスポットライトのテストである。4つのコイルはそれぞれ，スムーズ・シェーディングの有る／無し，マッピングの有る／無しの組合せと4通りのマテリアルの例を示している。スポットライトは7灯あり，すべて異なる色をもたせてある。また，光は光源からの距離により減衰させるタイプを使用している。ただし，指

向性はもたせていないので，コイルと同様に後ろの壁に光が映り込んでいるのがわかるであろう。

“Cat & Dog”は“Coils”をさらに複雑にし，2体のマリオネットを付け加えたもので，実際に.str データをどのように作るか参考にしてほしい。プリミティブとなる.obj のほとんどは makeprim の回転体で作ったものであるが，カーテン，サッカーボールなどいくつかは別の特殊なプログラムで作成している。しかし，これらも基本的には同じであり，難しくないので各自いろいろ研究してみるとよい。マッピングデータとしては，スキャナから取り込んだものを1回目に定義した.rgb(.map) データに変換して使用しているが，スキャナのない方には無理であろう。また，debut ではマッピングのアンチ・エイリアシングを行っていないので，初めのうち比較的小さなモデルであるマリオネットへのマッピングが胡麻塩模様になってしまった。これを解決するため，マッピングデータに初めからわざとぼかすという画像処理を行った.rgb を使って計算してみたら，かなりうまくいった。点光線は各 light.obj の下に取り付けてあるので，ちょうどそれぞれの電球が光っているような効果を与えている。

debut シリーズを使用する際は，

① まず ver.0 でワイヤーで表示しチェックする

② ver.1 をクイック・シェーダーとして使用し，簡単なレンダリングテストを行う

③ これでうまくいくと思えば，ver.2 で時間をかけて本番レンダリングをする

という順序で行えばよいと思う。

debut プログラムの今後の発展

debut はデプス・バッファ・プログラムであるので，かなりいろいろな制約がある。このバージョンでは，

① アンチ・エイリアシングを行っていない

② マッピングのアンチ・エイリアシングを行っていない

③ 半透明のマテリアルが使えない

④ バンプマップが使えない

⑤ 描画できる画像サイズの制限

⑥ 処理速度

などの問題点がある。しかし，①，③ は a バッファにすることにより解決すべきものであり，⑤，⑥ はさらにそれをスキャンラインにすることにより解決されると思う。力のある人たちはそれらに関する文献を参考にして発展させてもらいたい。

デプス・バッファであることを利用した面白い機能として，2次曲面プリミティブをポリゴンと同様に使用できるということがある。これに関しては文献1)を参考にしてほしい。

さらに，静止画では物足りず，アニメーションを作りたい方のために付記しておく。

アニメーションを作るためには単純には.str データ内の各回転や平行移動パラメータの値をどんどん変化させていけばよいのである。そのためにキーとなる.str をいくつか決めておき，各パラメータをスプラインにより補間するという方法をとる。また，オブジェクトを.str で組み立てたり，キーフレームを決めたりするためには，ぜひともツリーエディタが欲しいところである。これはワイヤー・フレームで全体の構造がモニターに表示され，各パラメータを変化させると（できればリアルタイムで），それが実際にモニターに表示されてチェックできるようにしたものである。これは，現在では PC-98 レベルのマイクロコンピュータでもかなりのものが制作できると思う(実際に数年前，あるハッカーが PC-98 でそれを作ってしまったのを目撃したことがある)。また，キーフレームによるアニメーション・データができたら，その動きをワイヤー・フレームでモニターに表示できるアニメーション・チェッカーも欲しいものである。これも PC-98 レベルのマイクロコンピュータでも制作可能のようである。

これで今回のシリーズを終わりにしたいと思いますが，もし連載の中で不明の部分がありましたら PIXEL 編集部までお便りを下さい（もし質問が多くあるようであれば，また別の機会にまとめて執筆するかもしれません)。また，debut を使用して何か作品ができたら，それも編集部に送ってみてはどうでしょうか？出来が良ければ掲載してもらえるかも……。

それでは，debut シリーズをパブリック・ドメインとしておおいに活用してください。

＊バグレポート1：正確にはバグではないのだが，今回や第1回目に定義した各データ内のラベルの名前が実際のソース・プログラムの中で異なっているものがあるようなので，注意して直してほしい。

参考文献

1) 出渕：「Z-バッファとレイトレーシングの複合アルゴリズム」，第2回 NICOGRAPH 論文コンテスト論文集，pp.29-38，1986

表1 makeprim ソース・プログラム

```
/************************************/
/*    PRM.H                   */
/*    header for makeprim.c      */
/************************************/

#define MAXLINE 240
#define EQUAL 0
#define MAXCOORD 10000 /* max coordinates for one object */
#define MAXPOINT 1000  /* max points for one line */
#define MAXVERT  100   /* max vertice of one polygon */
#define PAI 3.141592
#define TRUE 1
#define FALSE 0

#define VERTEX   "VX"
#define MAPVERT  "VM"
#define SURFACE  "SF"
#define MAPSURF  "FM"
#define MATERIAL "MT"
#define LINE           "LN" /* line -- using in .obj for rotation shape */
#define DLM      ',' /* char !*/

typedef struct{
                float       u,
                            v;
                } Map2f;

typedef struct{
                float       x,
                            y;
                } Vector2f;

typedef struct{
                float       x,
                            y,
                            z;
                } Vector3f;

Vector3f        coord[MAXCOORD];
Vector2f        point[MAXPOINT];
Map2f           mapv[MAXVERT];

typedef struct{
                char        name[MAXLINE];
                int         prim_type;
                int         patch_fg;
                int         map_fg;
                char        matname[MAXLINE];
                char        mtlname[MAXLINE];
                int         du;
                int         dv;
                int         PointNum;
                int         VertNum;
                int         FaceNum;
        } Object;

Object object[1];
```

```c
/*
*   ***************************
*   *     MAKEPRIM.c        *
*   ***************************
*
*   Written by Ryoichiro Debuchi
*   (CATDIG -- is a brand name of all my works)
*   Date: Feb. 1989   c & p R. Debuchi
*
*   Function:
*   Making Object File(.obj) Using in Debut Software.
*   This program produces several kinds of primitive objects
*   which are used in DEBUT(Ver. 0-2).
*   This provides you 8 kinds of primitives -- Mesh, N-Polygon,
*   Cube, Rotation shape, Sphere, Cylinder(Corn), Coil and Spiral.
*   This Process is performed by inputting commands or data from keybord.
*
*/

#include <stdio.h>
#include <math.h>
#include "prm.h"

float   pai;
FILE    *fp, *fopen();
char    savename[MAXLINE];

main()
{
        pai = PAI / 180.0;
        object->prim_type = SelectObj();
        SetObjPara();
        AtachMat();
        if(object->prim_type == 2)
                SaveFilePoly();
        else if(object->prim_type == 3)
                SaveFileCube();
        else{
                object->VertNum = object->dv * (object->du + 1);
                SaveFileUV();
        }
        exit(1);
}

int     SelectObj()
{
        int     type;

        label1:
        fprintf(stderr, "SELECT OBJECT TYPE\n");
        fprintf(stderr, "Rectangle Mesh = 1\n");
        fprintf(stderr, "N-Polygon     = 2\n");
        fprintf(stderr, "Cube          = 3\n");
        fprintf(stderr, "Rotation Shape = 4\n");
        fprintf(stderr, "Sphere        = 5\n");
        fprintf(stderr, "Cylinder(Corn) = 6\n");
        fprintf(stderr, "Coil          = 7\n");
        fprintf(stderr, "Spiral        = 8\n");

        GetKeyI(&type, 3, "Object Type=<%d>");
        if(type < 1 || type > 8){
                fprintf(stderr, "Illegal Object Type!\n");
                goto label1;
        }
        return(type);
}

SetObjPara()
{
        char    save[MAXLINE];

        switch(object->prim_type)
        {
                case 1: MakeMesh();     break;
                case 2: MakePolygon();  break;
                case 3: MakeCube();     break;
                case 4: MakeRotation(); break;
                case 5: MakeSphere();   break;
                case 6: MakeCylinder(); break;
                case 7: MakeCoil(0);    break;
                case 8: MakeCoil(1);    break;
        }
}

AtachMat()
{
        char    save[MAXLINE];

        if(object->prim_type != 2)
                GetKeyS(save, "t",
                        "Which? Triangle or Rectangle Patch(t/r)=<%s>");
        else{
                GetKeyS(save, "y", "Use Triangle Patchs ?(y/n)=<%s>");
                if(strcmp(save, "n") == EQUAL)
                        object->patch_fg = TRUE;
                else
                        object->patch_fg = FALSE;
        }

        GetKeyS(save, "y", "Atach Material?(y/n)=<%s>");
        if(strcmp(save, "y") == EQUAL){
                GetKeyS(object->matname, "mat", "Input .mat File Name=<%s>");
                GetKeyS(object->mtlname, "col", "Input Material Name=<%s>");
                GetKeyS(save, "n", "Use Mapping?(y/n)=<%s>");
                if(strcmp(save, "y") == EQUAL) object->map_fg = TRUE;
                else                    object->map_fg = FALSE;
        }
}

MakeRotation()
{
        int     i, j, sum;
        float   an, b, v, r, a;

        fprintf(stderr, "MAKE ROTATION SHAPE\n");
        ReadLineFile();
        object->dv = object->PointNum;

        GetKeyI(&(object->du), 12, "Input Division Num=<%d>");
        GetKeyF(&an, 360.0, "Input Angle=<%g>");
        b = (float)an / (float)object->du;

        for(i = 0; i < object->dv; i++)
        {
                v = (point+i)->y;
                r = (point+i)->x;
                for(j = 0; j <= object->du; j++)
                {
                        sum = i * (object->du + 1) + j;
                        a = b * j * pai;
                        (coord+sum)->x = r * cos((double)a);
                        (coord+sum)->z = r * sin((double)a);
                        (coord+sum)->y = v;
                }
        }
}

ReadLineFile()
{
        char    name[MAXLINE], save[MAXLINE];
        char    tank[MAXLINE], line[MAXLINE];
        char    *p;
        int     i, n, m, vertn, cn, pn;
        int     vn[MAXLINE], vtn[MAXLINE];
        float   vp[3];
        Vector2f stackp[MAXPOINT];

        fprintf(stderr, "INPUT READ OBJ. FILE NAME\n");
        label2:
        GetKeyS(save, "default", "Read File Name=<%s>");
        sprintf(name, "%s.obj", save);
        fp = fopen( name, "r");
        if(fp == (FILE *)NULL)
        {
                fprintf(stderr, "Can't Open File %s\n", name);
                fprintf(stderr, "Input Again\n");
                goto label2;
        }
        n = m = pn = 0;
        while(1)
        {
                p = fgets(line, MAXLINE, fp);
                if(p == NULL)
                        break;
                sscanf(line, "%s", tank);
                if(strcmp(tank, VERTEX) == EQUAL){
                        if(n >= MAXPOINT){
                                fprintf(stderr,
                        "Vertex too Large (<= %d) (%s)\n", MAXPOINT, name);
                                exit(1);
                        }
                        sscanf(line, "%s %f %f %f", tank, vp, vp+1, vp+2);
                        (stackp+pn)->x = *vp * 100.0;
                        (stackp+pn)->y = *(vp+1) * 100.0;
                        pn += 1;
                }
                else if(strcmp(tank, LINE) == EQUAL){
                        if(m == 0){
                                vertn = GetFace(line, vn, vtn);
                                if(vertn > MAXPOINT){
                                        fprintf(stderr,
        "Line Vertice too Large (<=%d) (%d l line) (%s)\n", MAXPOINT, m, name);
                                        exit(1);
                                }
```

```
						object->PointNum = vertn;
						for(i = 0; i < vertn; i++)
						{
							cn = *(vn+i) - 1;
							(point+i)->x = (stackp+cn)->x;
							(point+i)->y = (stackp+cn)->y;
							fprintf(stderr,
				"point=%d %g %g\n", i, (stackp+cn)->x, (stackp+cn)->y);
						}
					}
					m += 1;
				}
		}
		fclose(fp);
}

MakeMesh()
{
		char		save[MAXLINE];
		int		i, j, dv, du, iax, sum;
		float		v, u, dlu, dlv, ilu, ilv;

		fprintf(stderr, "MAKE MESH RECTANGLE\n");
		GetKeyS(save, "x", "Input Axis(x/y/z)=<%s>");
		if(strcmp(save, "y") == EQUAL)	iax = 1;
		else if(strcmp(save, "z") == EQUAL) iax = 2;
		else				iax = 0;

		GetKeyF(&ilu, 100.0, "Input Length U=<%g>");
		GetKeyF(&ilv, 100.0, "Input Length V=<%g>");
		GetKeyI(&du, 10, "Input Div Num U=<%d>");
		if(du == 0) du = 1;
		GetKeyI(&dv, 10, "Input Div Num V=<%d>");
		if(dv == 0) dv = 1;

		object->dv = dv + 1;
		object->du = du;
		dlu = ilu / (float)du;
		dlv = ilv / (float)dv;

		switch(iax)
		{
			case 0:
				for(i = 0; i <= dv; i++)
				{
					v = -ilv / 2.0 + i * dlv;
					for(j = 0; j <= du; j++)
					{
						sum = i * (du + 1) + j;
						u = -ilu / 2.0 + j * dlu;
						(coord+sum)->x = 0.0;
						(coord+sum)->y = u;
						(coord+sum)->z = v;
					}
				}
				break;
			case 1:
				for(i = 0; i <= dv; i++)
				{
					v = -ilv / 2.0 + i * dlv;
					for( j = 0; j <= du; j++)
					{
						sum = i * (du + 1) + j;
						u = -ilu / 2.0 + j * dlu;
						(coord+sum)->x = v;
						(coord+sum)->y = 0.0;
						(coord+sum)->z = u;
					}
				}
				break;
			default:
				for(i = 0; i <= dv; i++)
				{
					v = -ilv / 2.0 + i * dlv;
					for( j = 0; j <= du; j++)
					{
						sum = i * (du + 1) + j;
						u = -ilu / 2.0 + j * dlu;
						(coord+sum)->x = u;
						(coord+sum)->y = v;
						(coord+sum)->z = 0.0;
					}
				}
				break;
		}
		return(1);
}

MakeSphere()
{
		int		i, j, sum;
		float		s, a, b, r, anglu, anglv;
		float		agu, agv, ir;
		Vector3f		sca;

		fprintf(stderr, "MAKE SPHERE\n");
		GetKeyF(&ir, 100.0, "Input Radius R=<%g>");
		GetKeyF(&(sca.x), 1.0, "Input Scale X=<%g>");
		GetKeyF(&(sca.y), 1.0, "Input Scale Y=<%g>");
		GetKeyF(&(sca.z), 1.0, "Input Scale Z=<%g>");
		GetKeyI(&(object->du), 12, "Input Div Num U=<%d>");
		if(object->du == 0) object->du = 1;

		GetKeyI(&(object->dv), 6, "Input Div Num V=<%d>");
		if(object->dv == 0) object->dv = 1;

		GetKeyF(&agu, 360.0, "Input Angle U=<%g>");
		GetKeyF(&agv, 180.0, "Input Angle V=<%g>");

		anglu = (float)agu / (float)object->du;
		anglv = (float)agv / (float)object->dv;
		r = ir;

		for(i = 0; i <= object->dv; i++)
		{
			b = i * anglv * pai;
			s = sin( (double)b);
			for(j = 0; j <= object->du; j++)
			{
				sum = i * (object->du + 1) + j;
				a = j * anglu * pai;
				(coord+sum)->x = r * s * sca.x * cos((double)a);
				(coord+sum)->z = r * s * sca.z * sin((double)a);
				(coord+sum)->y = r * sca.y * cos((double)b);
			}
		}
		object->dv += 1;
		return(1);
}

MakeCube()
{
		int		i, j, sum;
		float		s, a, b, r, anglu, anglv;
		float		agu, agv, ir;
		Vector3f		sca;

		fprintf(stderr, "MAKE CUBE\n");
		GetKeyF(&(sca.x), 100.0, "Input Size X=<%g>");
		GetKeyF(&(sca.y), 100.0, "Input Size Y=<%g>");
		GetKeyF(&(sca.z), 100.0, "Input Size Z=<%g>");

		(coord)->x = 0.0;	(coord)->y = 0.0;	(coord)->z = 0.0;
		(coord+3)->x = 0.0;	(coord+3)->y = 0.0;	(coord+3)->z = sca.z;
		(coord+2)->x = sca.x; (coord+2)->y = 0.0;	(coord+2)->z = sca.z;
		(coord+1)->x = sca.x; (coord+1)->y = 0.0;	(coord+1)->z = 0.0;
		(coord+4)->x = 0.0;	(coord+4)->y = sca.y; (coord+4)->z = 0.0;
		(coord+7)->x = sca.x; (coord+7)->y = sca.y; (coord+7)->z = 0.0;
		(coord+6)->x = sca.x; (coord+6)->y = sca.y; (coord+6)->z = sca.z;
		(coord+5)->x = 0.0;	(coord+5)->y = sca.y; (coord+5)->z = sca.z;

		object->VertNum = 8;
		return(1);
}


MakeCylinder()
{
		char		save[MAXLINE];
		int		i, j, iax, sum;
		float		r1, r2, a, b, r, anglu;
		float		dv, dr, len, agu;
		Map2f		sca;

		fprintf(stderr, "MAKE CYLINDER\n");
		GetKeyS(save, "x", "Input Axis(x/y/z)=<%s>");

		if(strcmp(save, "y") == EQUAL)	iax = 1;
		else if(strcmp(save, "z") == EQUAL) iax = 2;
		else				iax = 0;

		GetKeyF(&r1, 100.0, "Input Radius1 R1=<%g>");
		GetKeyF(&r2, r1, "Input Radius2 R2=<%g>");
		GetKeyF(&(sca.u), 1.0, "Input Scale U=<%g>");
		GetKeyF(&(sca.v), 1.0, "Input Scale V=<%g>");
		GetKeyF(&len, 100.0, "Input Height H=<%g>");
		GetKeyI(&(object->du), 12, "Input Div Num U=<%d>");
		if(object->du == 0) object->du = 1;

		GetKeyI(&(object->dv), 6, "Input Div Num V=<%d>");
		if(object->dv == 0) object->dv = 1;

		GetKeyF(&agu, 360.0, "Input Angle U=<%g>");

		anglu = (float)agu / (float)object->du;
```

```
        dv = (float)len / (float)object->dv;

        if(object->dv != 0.0) dr = (r2 - r1) / (float)object->dv;
        else            dr = 0;

        for( i = 0; i <= object->dv; i++)
        {
                b = (object->dv - i) * dv;
                r = r1 + i * dr;
                for( j = 0; j <= object->du; j++)
                {
                        sum = i * (object->du + 1) + j;
                        a = j * anglu * pai;
                        switch(iax)
                        {
                                case 0:
                                (coord+sum)->z = r * sca.u * cos((double)a);
                                (coord+sum)->y = r * sca.v * sin((double)a);
                                (coord+sum)->x = b;
                                break;
                                case 1:
                                (coord+sum)->x = r * sca.u * cos((double)a);
                                (coord+sum)->z = r * sca.v * sin((double)a);
                                (coord+sum)->y = b;
                                break;
                                default:
                                (coord+sum)->y = r * sca.u * cos((double)a);
                                (coord+sum)->x = r * sca.v * sin((double)a);
                                (coord+sum)->z = b;
                                break;
                        }
                }
        }
        object->dv += 1;
        return(1);
}

MakePolygon()
{
        char    save[MAXLINE];
        int     i, j, iax, sum;
        float   a, b, r, anglu, dv, dr, len, agu;
        Map2f   sca;

        fprintf(stderr, "MAKE POLYGON\n");
        GetKeyS(save, "x", "Input Axis(x/y/z)=<%s>");

        if(strcmp(save, "y") == EQUAL)     iax = 1;
        else if(strcmp(save, "z") == EQUAL) iax = 2;
        else                        iax = 0;

        GetKeyF(&r, 100.0, "Input Radius R=<%g>");
        GetKeyF(&(sca.u), 1.0, "Input Scale U=<%g>");
        GetKeyF(&(sca.v), 1.0, "Input Scale V=<%g>");
        GetKeyF(&len, 0.0, "Input Height H=<%g>");
        GetKeyI(&(object->du), 12, "Input Div Num U=<%d>");

        anglu = 360.0 / (float)object->du;
        j = 0;
        switch(iax)
        {
                case 0:
                (coord+j)->z = 0; (coord+j)->y = 0; (coord+j)->x = len;
                break;
                case 1:
                (coord+j)->x = 0; (coord+j)->z = 0; (coord+j)->y = len;
                break;
                default:
                (coord+j)->y = 0; (coord+j)->x = 0; (coord+j)->z = len;
                break;
        }
        (mapv+j)->u = 0.5;
        (mapv+j)->v = 0.5;

        for(j = 1; j <= object->du; j++)
        {
                a = j * anglu * pai;
                switch(iax)
                {
                        case 0:
                        (coord+j)->z = r * sca.u * cos((double)a);
                        (coord+j)->y = r * sca.v * sin((double)a);
                        (coord+j)->x = len;
                        break;
                        case 1:
                        (coord+j)->x = r * sca.u * cos((double)a);
                        (coord+j)->z = r * sca.v * sin((double)a);
                        (coord+j)->y = len;
                        break;
                        default:
                        (coord+j)->y = r * sca.u * cos((double)a);
                        (coord+j)->x = r * sca.v * sin((double)a);
                        (coord+j)->z = len;
                        break;
                }
                (mapv+j)->u = cos((double)a) / 2.0 + 0.5;
                (mapv+j)->v = sin((double)a) / 2.0 + 0.5;
        }
        object->VertNum = object->du + 1;
        return(1);
}

MakeCoil(type)
int     type;
{
        int     i, j, sum;
        float   dh, dru, drv, a, b, r, ru, rv, anglu, anglv;
        float   agu, agv, iru, irv, ih;


        if(type == 0) fprintf(stderr, "MAKE COIL\n");
        else        fprintf(stderr, "MAKE SPIRAL\n");

        GetKeyF(&iru, 100.0, "Input Radius U=<%g>");
        GetKeyF(&irv, 20.0, "Input Radius V=<%g>");
        GetKeyI(&(object->du), 12, "Input Div Num U=<%d>");
        if(object->du == 0)
                object->du = 1;
        GetKeyI(&(object->dv), 12, "Input Div Num V=<%d>");
        if(object->dv == 0)
                object->dv = 1;
        GetKeyF(&agu, 360.0, "Input Angle U=<%g>");
        GetKeyF(&agv, 360.0, "Input Angle V=<%g>");
        GetKeyF(&ih, 0.0, "Input Height H=<%g>");

        anglu = (float)agu / (float)object->du;
        anglv = (float)agv / (float)object->dv;
        dh = (float)ih / (float)object->du;
        ru = iru;
        rv = irv;

        if(type == 0)
        {
                for(i = 0; i <= object->dv; i++)
                {
                        b = i * anglv * pai;
                        r = ru + rv * cos((double)b);
                        for(j = 0; j <= object->du; j++)
                        {
                                sum = i * (object->du + 1) + j;
                                a = j * anglu * pai;
                                (coord+sum)->x = r * cos((double)a);
                                (coord+sum)->y = r * sin((double)a);
                                (coord+sum)->z = rv * sin((double)b) + j * dh;
                        }
                }
        }
        else
        {
                dru = iru / (float)object->du;
                drv = irv / (float)object->du;
                for(i = 0; i <= object->dv; i++)
                {
                        for(j = 0; j <= object->du; j++)
                        {
                                sum = i * (object->du + 1) + j;

                                b = i * anglv * pai;
                                ru = j * dru;
                                rv = j * drv;
                                r = ru + rv * cos((double)b);
                                a = j * anglu * pai;
                                (coord+sum)->x = r * cos((double)a);
                                (coord+sum)->y = r * sin((double)a);
                                (coord+sum)->z = rv * sin((double)b) + j * dh;
                        }
                }
        }
        object->dv += 1;
        return(1);
}

OpenSaveFile()
{
        char    save[MAXLINE];

        fprintf(stderr, "INPUT SAVE OBJECT FILE NAME\n");

        label3:
        GetKeyS(save, "default", "Save File Name=<%s>");

        sprintf(savename, "%s.obj", save);
        fp = fopen(savename, "w");
        if(fp == (FILE *)NULL){
```

```
			fprintf(stderr, "Can't Open File %s\n", savename);
			fprintf(stderr, "Input Again\n");
			goto label3;
		}
}
SaveFileUV() /* for u-v type .obj */
{
	int	i, j, n, m, k;
	int	sp0, sp1, sp2, sp3;
	float	du, dv, dut, dvt;

	OpenSaveFile();

	fprintf(fp, "#MATRIX %d %d\n", object->dv, object->du);
	fprintf(fp, "# %s\n", savename);
	fprintf(fp, "# \n");

	for(i = 0; i < object->VertNum; i++)
		fprintf(fp, "%s %f %f %f\n",
   VERTEX, (coord+i)->x / 100.0, (coord+i)->y / 100.0, (coord+i)->z / 100.0);
	fprintf(fp, "# %d vertices\n", object->dv * (object->du + 1));
	fprintf(fp, "\n");

	if(object->map_fg == TRUE){
		du = 1.0 / (float)object->du;
		dv = 1.0 / (float)(object->dv - 1);
		for(j = 0; j < object->dv; j++)
		{
			dvt = j * dv;
			for(i = 0; i <= object->du; i++)
			{
				dut = i * du;
			    fprintf(fp, "%s %f %f 0.0\n", MAPVERT, dut, dvt);
			}
		}
		fprintf(fp, "# %d mapping vertices\n",
						object->dv * (object->du + 1));
	}
	fprintf(fp, "# \n");
	fprintf(fp, "%s %s %s\n", MATERIAL, object->matname, object->mtlname);
	m = object->dv - 1;
	for(j = 0; j < m; j++)
	{
		n = j * (object->du + 1) + 1;
		for(i = 0; i < object->du; i++)
		{
			k = n + i;
			sp0 = k;
			sp3 = k + object->du + 1;
			sp1 = sp0 + 1;
			sp2 = sp3 + 1;

			if(object->patch_fg == FALSE){
				if(object->map_fg == FALSE){
					fprintf(fp, "%s %d %d %d\n",
						SURFACE, sp0, sp1, sp2);
					fprintf(fp, "%s %d %d %d\n",
						SURFACE, sp0, sp2, sp3);
				}
				else{
					fprintf(fp, "%s %d%c%d %d%c%d %d%c%d\n",
 MAPSURF, sp0, DLM, sp0, sp1, DLM, sp1, sp2, DLM, sp2);
					fprintf(fp, "%s %d%c%d %d%c%d %d%c%d\n",
 MAPSURF, sp0, DLM, sp0, sp2, DLM, sp2, sp3, DLM, sp3);
				}
			}
			else{
				if(object->map_fg == FALSE)
				fprintf(fp, "%s %d %d %d %d\n",
						SURFACE, sp0, sp1, sp2, sp3);
				else
				fprintf(fp, "%s %d%c%d %d%c%d %d%c%d %d%c%d\n",
	MAPSURF, sp0, DLM, sp0,  sp1, DLM, sp1, sp2, DLM, sp2, sp3, DLM, sp3);
			}
		}
	}
	if(object->patch_fg == 0)
		fprintf(fp, "# %d elements\n",
					(object->dv - 1) * object->du * 2);
	else
		fprintf(fp, "# %d elements\n",
					(object->dv - 1) * object->du);
	fclose(fp);
	return(1);
}

SaveFileCube() /* for cube type .obj */
{
	int	i, j, n, m, k;
	int	sp0, sp1, sp2, sp3;
	float	du, dv, dut, dvt;

	OpenSaveFile();

	fprintf(fp, "# CUBE\n");
	fprintf(fp, "# %s\n", savename);
	fprintf(fp, "# \n");

	for(i = 0; i < object->VertNum; i++)
		fprintf(fp, "%s %f %f %f\n",
 VERTEX, (coord+i)->x / 100.0, (coord+i)->y / 100.0, (coord+i)->z / 100.0);
	fprintf(fp, "# %d vertices\n", 8);
	fprintf(fp, "\n");

	if(object->map_fg == TRUE){
		fprintf(fp, "%s 0.0 0.0 0.0\n", MAPVERT);
		fprintf(fp, "%s 0.0 1.0 0.0\n", MAPVERT);
		fprintf(fp, "%s 1.0 1.0 0.0\n", MAPVERT);
		fprintf(fp, "%s 1.0 0.0 0.0\n", MAPVERT);
		fprintf(fp, "# %d mapping vertices\n", 4);
	}
	fprintf(fp, "# \n");
	fprintf(fp, "%s %s %s\n", MATERIAL, object->matname, object->mtlname);
	if(object->map_fg == FALSE){
		fprintf(fp, "%s 1 2 3 4\n", SURFACE);
		fprintf(fp, "%s 5 6 7 8\n", SURFACE);
		fprintf(fp, "%s 1 5 8 2\n", SURFACE);
		fprintf(fp, "%s 2 8 7 3\n", SURFACE);
		fprintf(fp, "%s 3 7 6 4\n", SURFACE);
		fprintf(fp, "%s 4 6 5 1\n", SURFACE);
	}
	else{
		fprintf(fp, "%s 1%c1 2%c2 3%c3 4%c4\n",
						SURFACE, DLM, DLM, DLM, DLM);
		fprintf(fp, "%s 5%c1 6%c2 7%c3 8%c4\n",
						SURFACE, DLM, DLM, DLM, DLM);
		fprintf(fp, "%s 1%c1 5%c2 8%c3 2%c4\n",
						SURFACE, DLM, DLM, DLM, DLM);
		fprintf(fp, "%s 2%c1 8%c2 7%c3 3%c4\n",
						SURFACE, DLM, DLM, DLM, DLM);
		fprintf(fp, "%s 3%c1 7%c2 6%c3 4%c4\n",
						SURFACE, DLM, DLM, DLM, DLM);
		fprintf(fp, "%s 4%c1 6%c2 5%c3 1%c4\n",
						SURFACE, DLM, DLM, DLM, DLM);
	}
	fprintf(fp, "# %d elements\n", 6);
	fclose(fp);
	return(1);
}

SaveFilePoly() /* for n-polygon type .obj */
{
	int	i, j, n, m, k, n2;
	int	sp0, sp1, sp2, sp3;
	float	du, dv, dut, dvt;

	OpenSaveFile();

	fprintf(fp, "# N-POLY\n");
	fprintf(fp, "# %s\n", savename);
	fprintf(fp, "# \n");

	if(object->patch_fg == FALSE){
		for(i = 0; i < object->VertNum; i++)
			fprintf(fp, "%s %f %f %f\n",
VERTEX, (coord+i)->x / 100.0, (coord+i)->y / 100.0, (coord+i)->z / 100.0);
		fprintf(fp, "# %d vertices\n", object->VertNum);
	}
	else{
		for(i = 1; i < object->VertNum; i++)
			fprintf(fp, "%s %f %f %f\n",
VERTEX, (coord+i)->x / 100.0, (coord+i)->y / 100.0, (coord+i)->z / 100.0);
		fprintf(fp, "# %d vertices\n", object->VertNum - 1);
	}
	fprintf(fp, "\n");
	if(object->map_fg == TRUE){
		if(object->patch_fg == FALSE){
			for(i = 0; i < object->VertNum; i++)
				fprintf(fp, "%s %f %f %f\n",
				MAPVERT, (mapv+i)->u, (mapv+i)->v, 0.0);
			fprintf(fp, "# %d mapping vertices\n", object->VertNum);
		}
		else{
			for(i = 1; i < object->VertNum; i++)
				fprintf(fp, "%s %f %f %f\n",
				MAPVERT, (mapv+i)->u, (mapv+i)->v, 0.0);
			fprintf(fp, "# %d mapping vertices\n",
						object->VertNum - 1);
		}
	}
	fprintf(fp, "# \n");
	fprintf(fp, "%s %s %s\n", MATERIAL, object->matname, object->mtlname);
	if(object->patch_fg == TRUE){
		if(object->map_fg == FALSE){
```

```
				fprintf(fp, "%s ", SURFACE);
				for(i = 1; i < object->VertNum; i++)
					fprintf(fp, "%d ", i);
				fprintf(fp, "\n");
			}
			else{
				fprintf(fp, "%s ", MAPSURF);
				for(i = 1; i < object->VertNum; i++)
					fprintf(fp, "%d%c%d ", i, DLM, i);
				fprintf(fp, "\n");
			}
			fprintf(fp, "# %d elements\n", 1);
		}
		else{
			n = object->VertNum - 1;
			if(object->map_fg == FALSE){
				for(i = 1; i <= n; i++)
				{
					sp0 = 1;
					sp1 = i + 1;
					if(i != n) sp2 = sp1 + 1;
					else    sp2 = 2;
					fprintf(fp, "%s %d %d %d\n",
							SURFACE, sp0, sp1, sp2);
				}
			}
			else{
				for(i = 1; i <= n; i++)
				{
					sp0 = 1;
					sp1 = i + 1;
					if(i != n) sp2 = sp1 + 1;
					else    sp2 = 2;
					fprintf(fp, "%s %d%c%d %d%c%d %d%c%d\n",
				MAPSURF, sp0, DLM, sp0, sp1, DLM, sp1, sp2, DLM, sp2);
				}
			}
			fprintf(fp, "# %d elements\n", object->VertNum - 1);
		}
		fclose(fp);
		return(1);
}

/*
		General input routines
*/

#define	LINEBUF256
static	char	buf[LINEBUF];
static	char	buf1[LINEBUF];

int	GetKeyI(retval, def, prompt)
char	*prompt;
int	def;
int	*retval;
{
	while(1)
	{
		fprintf(stderr, prompt, def);
		fgets(buf, LINEBUF, stdin);
		if(*buf == '\n'){
			*retval = def;
			return(2);
		}
		if(sscanf(buf, "%d", retval) == 1)
			return(1);
	}
}
int	GetKeyC(retval, def, prompt)
char	*prompt;
char	def;
char	*retval;
{
	int	tmp;

	while(1)
	{
		fprintf(stderr, prompt, def);
		fgets(buf, LINEBUF, stdin);
		if(*buf == '\n'){
			*retval = def;
			return(2);
		}
		if(sscanf(buf, "%c", &tmp) == 1){
			*retval = (char)tmp;
			return(1);
		}
	}
}

int	GetKeyF(retval, def, prompt)
char	*prompt;
float	def;
float	*retval;
{
	while(1)
	{
		fprintf(stderr, prompt, def);
		fgets(buf, LINEBUF, stdin);
		if(*buf == '\n'){
			*retval = def;
			return(2);
		}
		if(sscanf(buf, "%f", retval) == 1)
			return(1);
	}
}

int	GetKeyS(retval, def, prompt)
char	*prompt;
char	*def;
char	*retval;
{
	while(1)
	{
		fprintf(stderr, prompt, def);
		fgets(buf, LINEBUF, stdin);
		if(*buf == '\n'){
			strcpy(retval, def);
			return(2);
		}
		if(sscanf(buf, "%s", retval) == 1)
			return(1);
	}
}

GetFace(tank, v, vt)
char	tank[MAXLINE];
int	v[MAXVERT], vt[MAXVERT];
{
	int	k, i, p, n, err;
	char	v_tank[MAXLINE], vt_tank[MAXLINE];

	p = err = 0;
	i = 2;
	while(1)
	{
		for( ; tank[i] == ' ' || tank[i] == '\t'; i++)
			;
		if(tank[i] == '\n')
			break;
		else{
			sprintf(v_tank, "          ");
			for(k = 0 ; tank[i] >= '0' && tank[i] <= '9'; i++, k++)
				v_tank[k] = tank[i];
				v[p] = iatoi(v_tank);

			if(tank[i] == DLM)
			{
				i++;
				sprintf(vt_tank, "          ");
			for(k = 0; tank[i] >= '0' && tank[i] <= '9'; i++, k++)

					vt_tank[k] = tank[i];
					vt[p] = iatoi(vt_tank);
			}
		}
		p++;
	}
	if(err == 0) n = p;
	else	n = -1;
	return(n);
}

iatoi( s)
char s[];
{
	int	i, n, sign;

	for( i = 0; s[ i] == ' ' || s[ i] == '\n' || s[ i] == '\t'; i++)
		;
	sign = 1;
	if( s[ i] == '+' || s[ i] == '-')
		sign = ( s[ i++] == '+') ? 1 : -1;
	for( n = 0; s[ i] >= '0' && s[ i] <= '9'; i++)
		n = 10 * n + s[ i] - '0';
	return( sign * n);
}
```

ラスター・グラフィックスのソフトウエア(28)

レイ・トレーシング(6)

前回，曲面パッチをレイ・トレーシング法で描くにあたって問題となった高次代数方程式の解き方を考え，そのプログラムを示す。

大野　義夫*

はじめに

曲面パッチをレイ・トレーシング法によって描くための Kajiya の方法[2]の概要を前回(89年2月号)説明した。Kajiya の方法では，光線と曲面パッチとの交点を求める計算を，終結式を利用することにより，18次方程式

$$p_{18}(z)=a_0z^{18}+a_1z^{17}+\cdots+a_{17}z+a_{18}=0 \quad (1)$$

を解く問題に帰着させる。Kajiya はこれをラゲール (Laguerre) の方法で解いた。ラゲールの方法も良い方法であるが，基本的に解を1個ずつ求めていく方法であり，われわれの問題の場合には0と1の間の実根のうちで始点に最も近い点に対応する解が必要である。隣接するピクセルの場合には，そのピクセルの位置に見える曲面の状況はあまり変わらないであろうから，前のピクセルで求まった解が今度のピクセルでも解を求めるための非常によい初期値となることはたいていの場合に間違いない。ところが，**図1**のような状況では，前のピクセルでは存在しなかった近い交点が今度のピクセルのところに突然現れることになる。前のピクセルでの解を初期値としたのでは，最も

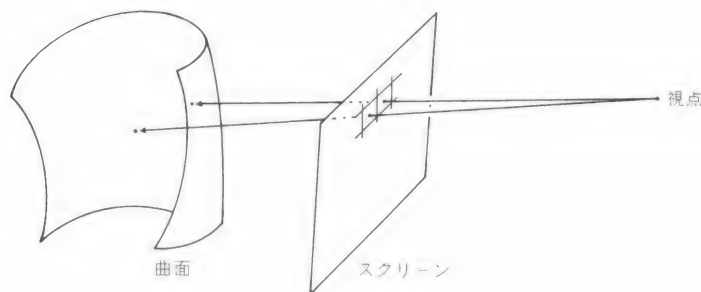



図1　隣接するピクセルと曲面との交点

* おおの　よしお　　慶応義塾大学　理工学部　電気工学科　〒223　神奈川県横浜市港北区日吉 3-14-1

近い交点ではなく，その次の交点が求まってしまうおそれが強い。こうしたことを考えると，結局は全部の解を求め，その中から必要なものを選択するといったかたちにせざるを得ない。全部の解を求める場合にも，前回述べたようにラゲールの方法が使えるが，1個の解が求まるたびに割り算をするために誤差が累積することと，プログラムが複雑になるという問題がある。

高次方程式のすべての解を同時に求め，数値的にも安定（どのような初期値から出発してもきちんと求める解に到達する）な解法としてDurand-Kerner法やAberth法などの方法がある。これらの方法についてはいくつかの解説もでているので(例えば文献1)や4)などを参照していただきたい)，ここでは，プログラムが読める程度に手順を示すだけにとどめておこう。

ニュートン法

方程式

$$f(x)=0 \tag{2}$$

の解を求めるにはニュートン法，あるいはニュートン-ラフソン法とよばれる方法が有名である。この方法は，解 α に十分に近い値 $x^{(0)}$ から出発して，

$$x^{(\nu+1)}=x^{(\nu)}-\frac{f(x^{(\nu)})}{f'(x^{(\nu)})} \tag{3}$$

の繰返しによって $x^{(1)}$, $x^{(2)}$, …を求めると，これがはじめの方程式の解に収束するというものである（残念ながら，常に収束するとはかぎらない）。

この方法は，連立方程式に拡張することができる。方程式

$$\begin{cases} f_1(x_1,\cdots,x_n)=0,\\ f_2(x_1,\cdots,x_n)=0,\\ \cdots\cdots\\ f_n(x_1,\cdots,x_n)=0 \end{cases} \tag{4}$$

すなわち

$$\boldsymbol{f}(\boldsymbol{x})=\boldsymbol{0} \tag{5}$$

に対しては，ヤコビ行列

$$J(x_1,\cdots,x_n)=[J_{ij}(x_1,\cdots,x_n)] =\begin{bmatrix} \frac{\partial f_1(\boldsymbol{x})}{\partial x_1} & \cdots & \frac{\partial f_1(\boldsymbol{x})}{\partial x_n}\\ \vdots & & \vdots\\ \frac{\partial f_n(\boldsymbol{x})}{\partial x_1} & \cdots & \frac{\partial f_n(\boldsymbol{x})}{\partial x_n} \end{bmatrix} \tag{6}$$

の逆行列

$$K(\boldsymbol{x})=[K_{ji}(\boldsymbol{x})]=[J_{ij}(\boldsymbol{x})]^{-1} \tag{7}$$

が求まれば，解 $\alpha=[\alpha_1,\cdots,\alpha_n]^t$ の近くの $\boldsymbol{x}^{(0)}=[x_1^{(0)},\cdots,x_n^{(0)}]^t$ から出発して

$$\begin{cases} x_1^{(\nu+1)}=x_1^{(\nu)}-\sum_{i=1}^{n}K_{ji}(\boldsymbol{x}^{(\nu)})f_i(\boldsymbol{x}^{(\nu)}),\\ \cdots\cdots\\ x_n^{(\nu+1)}=x_n^{(\nu)}-\sum_{i=1}^{n}K_{ji}(\boldsymbol{x}^{(\nu)})f_n(\boldsymbol{x}^{(\nu)}) \end{cases} \tag{8}$$

の繰返しによって得られる $\boldsymbol{x}^{(1)}$, $\boldsymbol{x}^{(2)}$, …が解 α に収束する（これも出発値の選び方によっては収束しないことがある）。

Aberth 法

代数方程式

$$p_n(z)=a_0z^n+a_1z^{n-1}+\cdots+a_{n-1}z+a_n =0 \quad (a_0\neq 0) \tag{9}$$

の解を $\alpha_1,\alpha_2,\cdots,\alpha_n$ とすると(もちろん，解は実数とはかぎらない。複素数も含まれる)，

$$p_n(z)=a_0(z-\alpha_1)(z-\alpha_2)\cdots(z-\alpha_n) \tag{10}$$

を展開して式(9)と係数を比較することにより，"解と係数の関係"

$$\begin{aligned} &\sum_{i=1}^{n}z_i=-\frac{a_1}{a_0},\\ &\sum_{i=1}^{n}\sum_{j=i+1}^{n}z_iz_j=\frac{a_2}{a_0},\\ &\cdots\\ &\sum_{i_1=1}^{n}\sum_{i_2=i_1+1}^{n}\cdots\sum_{i_k=i_{k-1}+1}^{n}z_{i_1}z_{i_2}\cdots z_{i_k} =(-1)^k\frac{a_k}{a_0},\\ &\cdots\\ &z_1z_2\cdots z_n=(-1)^n\frac{a_n}{a_0} \end{aligned} \tag{11}$$

が得られる。これを $z_1,z_2,\cdots,z_n$ を未知数とする連立方程式

$$\begin{aligned} &f_1(z_1,\cdots,z_n)=\sum_{i=1}^{n}z_i+\frac{a_1}{a_0}=0,\\ &f_2(z_1,\cdots,z_n)=\sum_{i=1}^{n}\sum_{j=i+1}^{n}z_iz_j-\frac{a_2}{a_0}=0,\\ &\cdots\\ &f_k(z_1,\cdots,z_n)=\sum_{i_1=1}^{n}\sum_{i_2=i_1+1}^{n}\cdots\sum_{i_k=i_{k-1}+1}^{n}z_{i_1}z_{i_2}\cdots z_{i_k} -(-1)^k\frac{a_k}{a_0}=0,\\ &\cdots\\ &f_n(z_1,\cdots,z_n)=z_1z_2\cdots z_n-(-1)^n\frac{a_n}{a_0}=0 \end{aligned} \tag{12}$$

として，前節のニュートン法によって解こうというのが，Durand-Kerner法やAberth法の原理である($z_1,z_2,\cdots,z_n$ はそれぞれ $\alpha_1,\alpha_2,\cdots,\alpha_n$ のいずれかを表すものとする)。ここでは，その中でも収束の速いAberth法を使ってみよう。

$$p_n(z)=a_0z^n+a_1z^{n-1}+\cdots+a_{n-1}z+a_n =a_0\prod_{i=1}^{n}(z-\alpha_i) \tag{13}$$

の対数をとると，

$$\log p_n(z)=\log a_0+\sum_{i=1}^{n}\log(z-\alpha_i) \tag{14}$$

が得られ，この両辺を z で微分すると，

$$\frac{p'_n(z)}{p_n(z)}=\sum_{i=1}^{n}\frac{1}{z-\alpha_i} =\frac{1}{z-\alpha_i}+\sum_{\substack{j=1,\cdots,n\\ \neq i}}\frac{1}{z-\alpha_j} \tag{15}$$

となる。さらに，

$$\begin{aligned} \frac{p'_n(z_i)}{p_n(z_i)}&=\frac{1}{z_i-\alpha_i}+\sum_{\substack{j=1,\cdots,n\\ \neq i}}\frac{1}{z_i-\alpha_j}\\ &=\frac{1}{z_i-\alpha_i}+\sum_{\substack{j=1,\cdots,n\\ \neq i}}\frac{z_i-z_j}{(z_i-z_j)(z_i-\alpha_j)}\\ &=\frac{1}{z_i-\alpha_i}+\sum_{\substack{j=1,\cdots,n\\ \neq i}}\frac{z_i-\alpha_j-z_j+\alpha_j}{(z_i-z_j)(z_i-\alpha_j)}\\ &=\frac{1}{z_i-\alpha_i} +\sum_{\substack{j=1,\cdots,n\\ \neq i}}\left[\frac{1}{z_i-z_j}-\frac{z_j-\alpha_j}{(z_i-z_j)(z_i-\alpha_j)}\right] \end{aligned} \tag{16}$$

から

$$\begin{aligned} \frac{1}{z_i-\alpha_i}&=\frac{p'_n(z_i)}{p_n(z_i)} -\sum_{\substack{j=1,\cdots,n\\ \neq i}}\left[\frac{1}{z_i-z_j}-\frac{z_j-\alpha_j}{(z_i-z_j)(z_i-\alpha_j)}\right]\\ &=\frac{p'_n(z_i)}{p_n(z_i)}-\sum_{\substack{j=1,\cdots,n\\ \neq i}}\frac{1}{z_i-z_j} +\sum_{\substack{j=1,\cdots,n\\ \neq i}}\frac{z_j-\alpha_j}{(z_i-z_j)(z_i-\alpha_j)} \end{aligned} \tag{17}$$

となって，

$$z_i-\alpha_i=\left[\frac{p'_n(z_i)}{p_n(z_i)}-\sum_{\substack{j=1,\cdots,n\\ \neq i}}\frac{1}{z_i-z_j} +\sum_{\substack{j=1,\cdots,n\\ \neq i}}\frac{z_j-\alpha_j}{(z_i-z_j)(z_i-\alpha_j)}\right]^{-1} \tag{18}$$

が得られる。z_i が α_i に十分に近いものとすると，

$$\sum_{\substack{j=1,\cdots,n\\ \neq i}}\frac{z_j-\alpha_j}{(z_i-z_j)(z_i-\alpha_j)} \tag{19}$$

が小さいことから，

$$\begin{aligned} \alpha_i&\approx z_i-\frac{1}{\dfrac{p'_n(z_i)}{p_n(z_i)}-\sum_{\substack{j=1,\cdots,n\\ \neq i}}\dfrac{1}{z_i-z_j}}\\ &\approx z_i-\frac{1\Big/\dfrac{p'_n(z_i)}{p_n(z_i)}}{1-\sum_{\substack{j=1,\cdots,n\\ \neq i}}\dfrac{1}{z_i-z_j}\Big/\dfrac{p'_n(z_i)}{p_n(z_i)}}\\ &\approx z_i-\frac{p_n(z_i)/p'_n(z_i)}{1-\dfrac{p_n(z_i)}{p'_n(z_i)}\sum_{\substack{j=1,\cdots,n\\ \neq i}}\dfrac{1}{z_i-z_j}} \end{aligned} \tag{20}$$

が得られる。このことから，次の繰返しによって解が求まることが期待できる。

```
  1: program aberth(input,output);
  2:
  3: const nmax=20; nmax1=21;
  4:
  5: type complex=record re,im:real
  6:              end;
  7:
  8: var a,aderiv:array[0..nmax] of real;
  9:     z:array[1..nmax] of complex;
 10:     n,i,isw:integer;
 11:
 12: function add(c1,c2:complex):complex;
 13: var w:complex;
 14: begin
 15:    w.re:=c1.re+c2.re;
 16:    w.im:=c1.im+c2.im;
 17:    add:=w
 18: end; { add }
 19:
 20: function sub(c1,c2:complex):complex;
 21: var w:complex;
 22: begin
 23:    w.re:=c1.re-c2.re;
 24:    w.im:=c1.im-c2.im;
 25:    sub:=w
 26: end; { sub }
 27:
 28: function mul(c1,c2:complex):complex;
 29: var w:complex;
 30: begin
 31:    w.re:=c1.re*c2.re-c1.im*c2.im;
 32:    w.im:=c1.re*c2.im+c1.im*c2.re;
 33:    mul:=w
 34: end; { mul }
 35:
 36: function dv(c1,c2:complex):complex;
 37: var ww:real;
 38:     w:complex;
 39: begin
 40:    ww:=sqr(c2.re)+sqr(c2.im);
 41:    w.re:=(c1.re*c2.re+c1.im*c2.im)/ww;
 42:    w.im:=(c1.im*c2.re-c1.re*c2.im)/ww;
 43:    dv:=w
 44: end; { dv }
 45:
 46: function p(zz:complex):complex;
 47: var w:complex;
 48:     i:integer;
 49: begin
 50:    w.re:=a[0];  w.im:=0.0;
 51:    for i:=1 to n do
 52:       begin w:=mul(w,zz);
 53:             w.re:=w.re+a[i]
 54:       end;
 55:    p:=w
 56: end; { p }
 57:
 58: function pderiv(zz:complex):complex;
 59: var w:complex;
 60:     i:integer;
 61: begin
 62:    w.re:=aderiv[0];  w.im:=0.0;
 63:    for i:=1 to n-1 do
 64:       begin w:=mul(w,zz);
 65:             w.re:=w.re+aderiv[i]
 66:       end;
 67:    pderiv:=w
 68: end; { pderiv }
 69:
 70: procedure initial;
 71: var b:array[0..nmax1,0..nmax] of real;
 72:     c:array[0..nmax] of real;
 73:     k,m,num:integer;
 74:     beta,r,rr,t,v1,v2:real;
 75: begin
 76:    beta:=-a[1]/(n*a[0]);
 77:    for k:=0 to n do b[0,k]:=a[k];
 78:    for m:=0 to n do
 79:       begin
 80:          b[m+1,0]:=a[0];
 81:          for k:=1 to n-m do
 82:             b[m+1,k]:=b[m+1,k-1]*beta+b[m,]
 83:       end;
 84:    num:=0;
 85:    for m:=0 to n-1 do
 86:       if b[m+1,n-m]<>0.0 then num:=num+1;
 87:    r:=0.0;
 88:    for k:=0 to n do
 89:       c[k]:=b[n-k+1,k];
 90:    for k:=1 to n do
 91:       begin
 92:          rr:=exp(ln(num*abs(c[k]/c[0]))/k)
 93:          if rr>r then r:=rr
 94:       end;
 95:    c[0]:=abs(c[0]);
 96:    for k:=1 to n do
 97:       c[k]:=-abs(c[k]);  writeln; writeln('r=        ',r);
 98:    for i:=1 to 2 do
 99:       begin
100:          v1:=c[0];
101:          for k:=1 to n do
102:             v1:=v1*r+c[k];
103:          v2:=n*c[0];
104:          for k:=1 to n-1 do
105:             v2:=v2*r+(n-k)*c[k];
106:          r:=r-v1/v2;  writeln('r becomes',r)
107:       end;
108:    for k:=1 to n do
109:       begin
110:          t:=(6.2832*(k-1)+1.5)/n;
111:          z[k].re:=beta+r*cos(t);
112:          z[k].im:=r*sin(t)
113:       end
114: end; { initial }
115:
116: procedure aberth;
117: const eps=1.0e-10; limit=50;
118: var istep,i:integer;
119:     sw,sw2:boolean;
120:     w:complex;
121:
122:    function psi(i:integer):complex;
123:    var unity,sum,ratio,w:complex;
124:        j:integer;
125:    begin
126:       unity.re:=1.0;  unity.im:=0.0;
127:       sum.re:=0.0;  sum.im:=0.0;
128:       for j:=1 to n do
129:          if j<>i
130:             then sum:=add(sum,dv(unity,sub(z[i],z[j])));
131:       ratio:=dv(p(z[i]),pderiv(z[i]));
132:       w:=dv(ratio,sub(unity,mul(ratio,sum)));
133:       w.re:=-w.re;  w.im:=-w.im;
134:       psi:=w
135:    end; { psi }
136:
137: begin
138:    istep:=0;
139:    writeln('nu=',istep:2);
140:    for i:=1 to n do
141:       writeln('  [',i:2,']',z[i].re:15:7,z[i].im:15:7);
142:    sw2:=false;
143:    while not sw2 do
144:    begin
145:       sw:=false;  i:=0;  istep:=istep+1;
146:       repeat
147:          i:=i+1;
148:          w:=p(z[i]);
149:          if sqr(w.re)+sqr(w.im)>eps then sw:=true
150:       until (i=n) or sw;
151:       if (not sw) or (istep>limit)
152:          then sw2:=true
153:          else
154:             begin
155:                for i:=1 to n do
156:                   z[i]:=add(z[i],psi(i));
157:                writeln('nu=',istep:2);
158:                for i:=1 to n do
159:                   writeln('  [',i:2,']',z[i].re:15:7,z[i].im:15:7
160:             end
161:    end
162: end; { aberth }
163:
164: begin { main }
165:    write('n:');
166:    read(n);
167:    write('a0,...,an:');
168:    for i:=0 to n do
169:       read(a[i]);
170:    for i:=0 to n-1 do
171:       aderiv[i]:=a[i]*(n-i);
172:    write('initial value -- Aberth(1)/Specify(0):');
173:    read(isw);
174:    if isw=0
175:       then
176:          for i:=1 to n do
177:             begin
178:                write('z0[',i:2,']:');
179:                read(z[i].re,z[i].im)
180:             end
181:       else initial;
182:    writeln;  writeln('coefficients:');
183:    for i:=0 to n do
184:       writeln('a[',i:2,']=',a[i]:12:2);
185:    aberth
186: end.
```

図 2 Aberth 法のプログラム

◉図 2 は前ページ。

$$\begin{cases} z_i^{(\nu+1)} = z_i^{(\nu)} + \psi_i(z_1^{(\nu)}, \cdots, z_n^{(\nu)}), \\ \psi_i(z_1^{(\nu)}, \cdots, z_n^{(\nu)}) \\ \quad = -\dfrac{p_n(z_i)/p'_n(z_i)}{1 - \dfrac{p_n(z_i)}{p'_n(z_i)} \displaystyle\sum_{\substack{j=1,\cdots,n \\ \neq i}} \dfrac{1}{z_i - z_j}} \end{cases} \qquad (21)$$

この方法が Aberth 法とよばれ，3 次の収束をする方法として知られている。前に名前を示した Durand-Kerner 法も似たような考え方の方法であり，収束は 2 次で多少遅いが，数値的により安定している。この他にも Aberth 法をさらに修正して，4 次以上の収束を行うようにした方法[3]も提案されている。

繰返しの初期値は，複素数さえ使えばたいていのもので収束するが，ここで示すプログラム例では，文献 1）(pp. 152-153) による“アバース (Aberth) の初期値を少し修正したもの”を用いるか，あるいは特定の初期値を指定できるようにしておいた。文献 1) による初期値は，複素平面上ですべての解の重心 $(\alpha_1 + \cdots + \alpha_n)/n$ を中心として（重心の位置 $[\beta, 0]$ は解と係数の関係から $\beta = -a_1/(na_0)$ と簡単に求められる)，すべての解を覆うようなあまり大きくない円を作り，その円周上で等間隔に複素数を選ぶものである。

プログラム

以上の手順をプログラムにしたものを**図 2** に示す。このプログラムは 20 次までの代数方程式を解く。ただし多少収束を速めるため，式 (21) を次のように変えて用いた。

$$\begin{aligned} z_1^{(\nu+1)} &= z_1^{(\nu)} + \psi_1(z_1^{(\nu)}, \cdots, z_n^{(\nu)}), \\ z_2^{(\nu+1)} &= z_2^{(\nu)} + \psi_2(z_1^{(\nu+1)}, z_2^{(\nu)}, \cdots, z_n^{(\nu)}), \\ z_3^{(\nu+1)} &= z_3^{(\nu)} + \psi_3(z_1^{(\nu+1)}, z_2^{(\nu+1)}, z_3^{(\nu)}, \cdots, z_n^{(\nu)}), \\ &\cdots\cdots \\ z_{n-1}^{(\nu+1)} &= z_{n-1}^{(\nu)} + \psi_{n-1}(z_1^{(\nu+1)}, \cdots, z_{n-2}^{(\nu+1)}, z_{n-1}^{(\nu)}, z_n^{(\nu)}), \\ z_n^{(\nu+1)} &= z_n^{(\nu)} + \psi_n(z_1^{(\nu+1)}, \cdots, z_{n-1}^{(\nu+1)}, z_n^{(\nu)}) \end{aligned} \qquad (22)$$

与えられた n 次方程式を $p_n(z)=0$ としたとき，z の n 次多項式 $p_n(z)$ の係数を配列 a に入れ，$p'_n(z)$ の係数を aderiv に入れる。また，解の近似値 $z_1^{(\nu)}, \cdots, z_n^{(\nu)}$ を配列 z に入れる。式 (21) の上のように変形したので，もう 1 回繰返しを進めたときの近似値 $z_i^{(\nu+1)}$ を別の配列に入れる必要はなく，そのまま配列 z のはじめの方から順に入れていってしまってかまわない。

12 行目から 44 行目までは 2 つの複素数 c1，c2 の加減乗除を行うための関数 add，sub，mul，dv の定義である*。また，46〜56 行目の関数 p は，

$$p_n(z) = a_0 z^n + a_1 z^{n-1} + \cdots + a_{n-1} z + a_n \qquad (23)$$

の値を求め，58〜68 行目の関数 pderiv は

$$p'_n(z) = na_0 z^{n-1} + (n-1) a_1 z^{n-2} + \cdots + a_{n-1} \qquad (24)$$

の値を求める。

＊標準的な Pascal では，関数の値となり得るのはスカラー値だけで，このプログラムのようにレコードを返すことはできない。ここではそのようにするとプログラムが煩雑になってわかりにくくなるので，そのまま complex 型というレコードの値を返すように書いてある。

70〜114 行目の手続き initial は文献 1）による初期値を決定する。この部分についての詳細は文献 1）の 152〜154 ページを参照していただきたい。基本的には，すべての解を覆う円の中心 β を 76 行目で求め，半径 r を 90〜94 行目のループで求めている。この r はかなり大き目なので，98〜107 行目のループでもう少し改良した小さい値に変更する。最後にこの円の円周上に等間隔で並ぶ複素数を初期値として配列 z に入れるのが 108〜113 行目である。

116〜165 行目が（3 次収束をする）Aberth 法の本体部分である。これをこのプログラムでは手続き aberth としてある。この中の 122〜135 行目は，解の近似値に加える修正量 $\psi_i(z_1, \cdots, z_n)$ を計算する関数 psi が占めている。ψ_i の計算は式 (21) そのままである。

Aberth 法の繰返しは 144〜161 行目のループである。繰返しを 1 回行うたびに istep の値を 1 ずつ増やす。ここでは istep の値が limit (＝50) を超えたら，あきらめて繰返しをやめるようにしてある。146〜150 行目のループで解に収束したかどうかを判定し，まだであれば解の修正値 ψ_i を求めて各 z_i に加える (154〜160 行目)。

主プログラム（164〜186 行目）では，必要なデータ $n, a_0, a_1, \cdots, a_n$ を読み込んだ後，文献 1）による初期値を使用するのか，別に初期値を指定するのかを尋ねる。前者であれば initial を呼び，後者であれば初期値（の実部と虚部）を読み込む。いずれにしても，その後 185 行目で aberth を呼ぶ。

例

Aberth 法を実際にいくつかの方程式に適用してみよう。

■例 1 ： 3 次方程式

簡単な 3 次方程式として $z^3-1=0$ を試してみる。この解はもちろん 1 と $(-1 \pm \sqrt{3}i)/2$ である。初期値として文献 1) によるものを用いた場合の計算経過を**図 3** に示し，これを複素平面上にプロットしたものを**図 4** に示す。この図でやや大きな白丸が初期値を示し，やや大きな黒丸が得られた解を示す。また初期値として任意に，例えば $1+i, 1+2i, 1-i$ を指定した場合の経過を**図 5** に示す。

	実部	虚部
nu= 0		
[1]	0.8775826	0.4794255
[2]	-0.8539885	0.5202918
[3]	-0.0235868	-0.9997218
nu= 1		
[1]	0.9947838	-0.1020062
[2]	-0.4596238	0.8881137
[3]	-0.4865912	-0.8736298
nu= 2		
[1]	1.0000000	0.0000960
[2]	-0.4999908	0.8660307
[3]	-0.5000000	-0.8660254
nu= 3		
[1]	1.0000000	0.0000000
[2]	-0.5000000	0.8660254
[3]	-0.5000000	-0.8660254

nu は繰返しの回数，[ ] は Z の添字を示す。

図 3　方程式 $Z^3-1=0$ に対する計算経過で，初期値は文献 1) による方法で決定

	実部	虚部
[1]	6.5917946	0.5088279
[2]	6.0503829	2.5616629
[3]	4.8395101	4.3055220
[4]	3.1052258	5.5300692
[5]	1.0567115	6.0876054
[6]	-1.0589509	5.9108832
[7]	-2.9865798	5.0212178
[8]	-4.4936739	3.5259167
[9]	-5.3984543	1.6053361
[10]	-5.5917909	-0.5088727
[11]	-5.0503641	-2.5617036
[12]	-3.8394785	-4.3055539
[13]	-2.1051852	-5.5300884
[14]	-0.0566667	-6.0876095
[15]	2.0589943	-5.9108717
[16]	3.9866167	-5.0211922
[17]	5.4936998	-3.5258801
[18]	6.3984661	-1.6052928

図 7　18 次方程式 (25) を解くための初期値

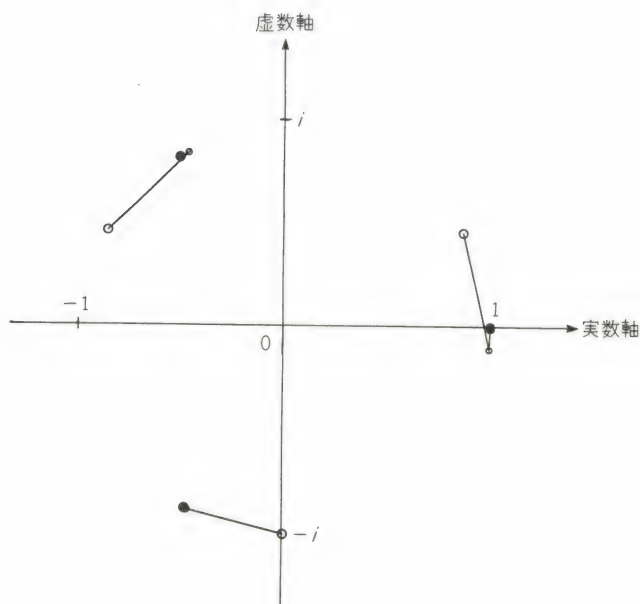



図4　図3をプロットした結果

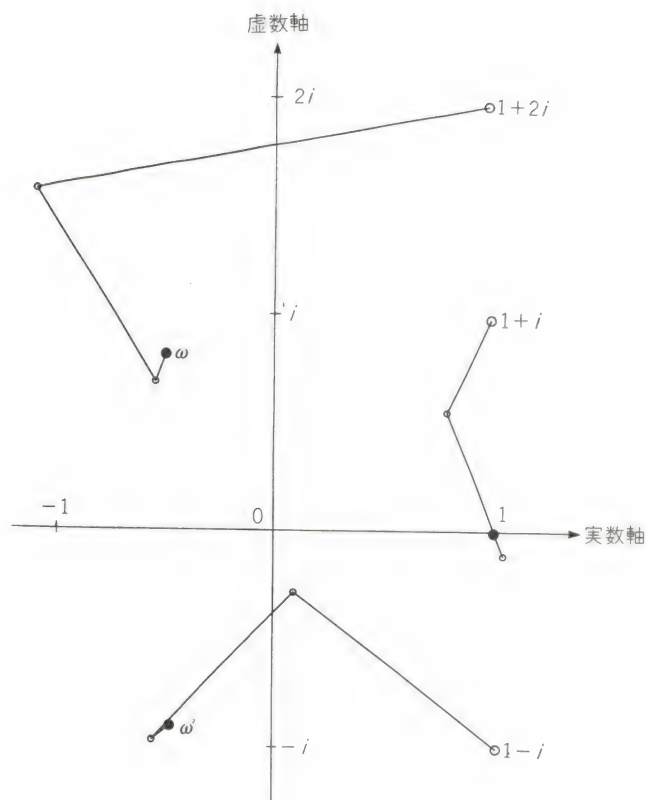



図5　方程式 $Z^3-1=0$ で初期値を $1+K_i$, $i=-1, 1, 2$ とした場合の計算

◉図8は次ページ。

■例2：18次方程式

図6 に示すような解を18個適当に定める。この解を持つ方程式 $p_{18}(z)=0$ は次のように得られる。

$$\begin{aligned}p_{18}(z)=&(z-3)(z-2)(z-1)(z+1)(z^2+1)(z^2+4)\\&[(z-2)^2+1][(z-1)^2+1][(z-1)^2+4]\\&[(z+1)^2+1][(z+1)^2+4]\\=&z^{18}-9z^{17}+41z^{16}-139z^{15}+392z^{14}-958z^{13}\\&+2,042z^{12}-3,718z^{11}+5,733z^{10}-7,657z^9\\&+8,593z^8-9,347z^7+12,154z^6\\&-10,976z^5+1,324z^4\\&+13,204z^3-18,280z^2+19,600z-12,000 \quad (25)\end{aligned}$$

これを解くための文献1）の初期値は**図7**のようになり，さらにAberth法の計算経過は**図8**のようになって，すべての解が正しく得られた。

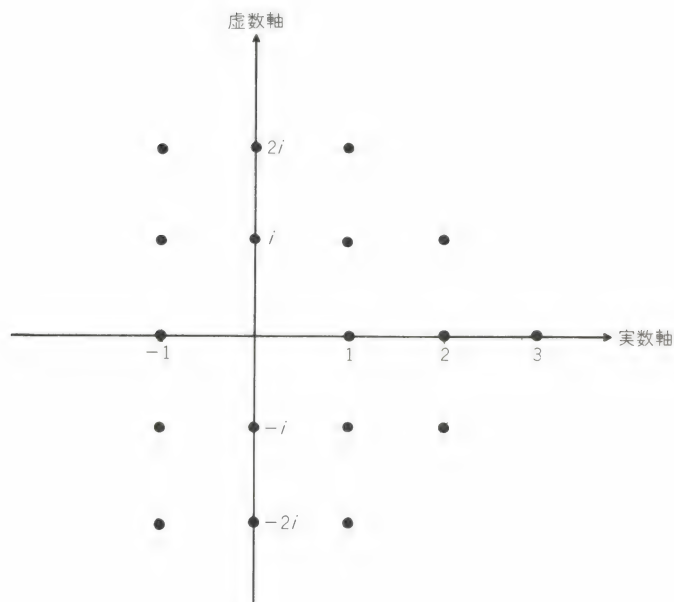



図6　適当に定めた18個の解

おわりに

前回のKajiyaの方法の中で，高次代数方程式を解く部分に着目し，扱いの簡単なAberth法に置き換えてみた。その結果，今回はあまりコンピュータ・グラフィックスらしい話題がでなかった。プ

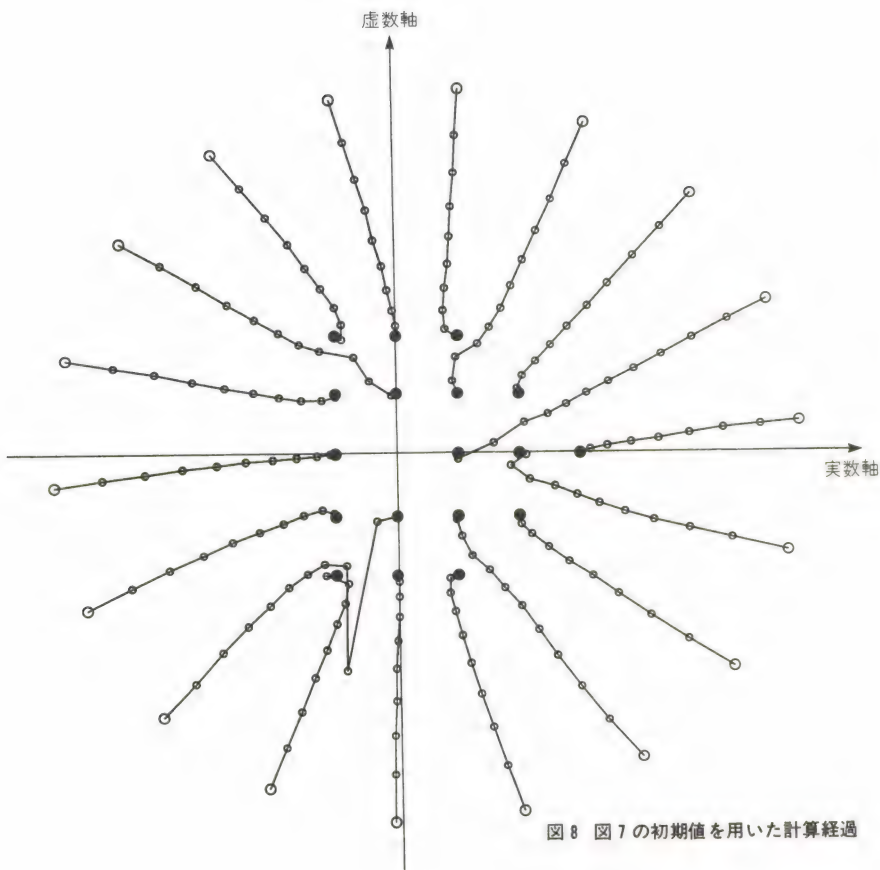



図8 図7の初期値を用いた計算経過

ログラムもAberth法の実験を行うためだけのものであった。次回は今回の結果をレイ・トレーシングと結合することを考えてみたい。

謝辞
高次代数方程式の解法についていろいろお教えいただいた，慶応義塾大学理工学部数理科学科の野寺隆さんに感謝いたします。

参考文献

1) 伊理正夫：「数値計算」，理工系基礎の数学12，朝倉書店，1981
2) J. T. Kajiya："Ray Tracing Parametric Patches", SIGGRAPH '82 Proceedings, pp. 245-254
3) 野寺隆：「4次収束をする代数方程式の解法」，情報処理学会第27回（昭和58年後期）全国大会講演論文集1F-3, pp. 1255-1256, 1983
4) 山本哲郎：「ある代数方程式解法と解の事後評価法」，数理科学，pp. 52-57, 1976. 7

BIBLIOGRAPHY

The Visual Computer

Vol.4, No.5 Nov. 1988

◇メカトロニクス・エンジニアリング教育・普及プログラム
◇超越関数（$z \rightarrow \cosh(z)+\mu$）のもつ興味深いグラフィック特性
◇デジタル直線の平行，重複，交差
◇宝石の炎：多面体の適応分散レイ・トレーシング
◇カオス理論を用いてレンダリング・マップで画像処理を行うための技法

Vol.4, No.6 Dec. 1988

機械工学，制御，アニメーション特集号

◇動力学アニメーション：相互作用と制御
◇ダイナミックな物体のアニメーションにおける複雑な運動拘束のための手法
◇変形モデル
◇単純な法則から複雑な動きを得る物理法則シミュレーション

問合せ先 Springer-Verlag Berlin Heidelberg Tokyo, Hongo 3-chome, Bunkyo-ku, Tokyo 113, Japan

連載

CGのための図学(13)

コンピュータ・グラフィックスで扱われる図学――図形処理の基礎学について，プログラムリストを示しながら説明する。

長島　忍*

はじめに

この連載は1988年4月号に始まり，当初1年間の予定であったが，都合によりもうしばらく延長させていただくことになった。

前回までワイヤー・フレームによる表示を扱ってきたが，今回はPascalの動的データを用いた図形のデータ構造について考えてみたい。例題として少し幾何模様的な図形の描画を行う。

使用システム

連載が始まって約1年経過したので，使用しているソフト／ハード・システムおよびグラフィック・ライブラリについて説明しておく。

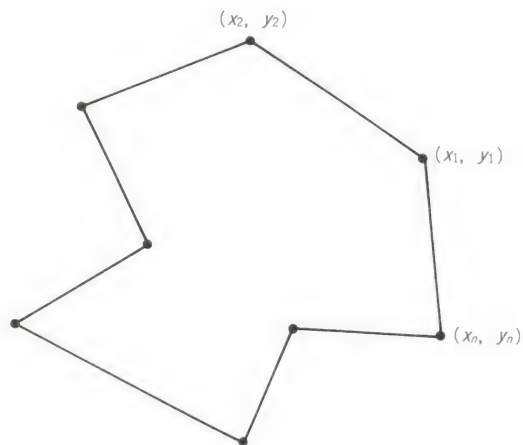
使用しているシステムは，NECのPC-9801VXとTurbo PASCAL Ver.3である。また，グラフィックス・ライブラリを**図1**に示す。このライブラリの機能はソース・リストの先頭に書いているように，初期化，画面消去，線分描画，線分の色変更，ハードコピーだけである。図形を表示するプログラムの先頭でこのライブラリのファイル名，例えばPLOT.LIBをインクルードする命令を入れて利用する。

ハードコピーにはこれまで，富士通のFMPR-351のプリンタを用いてきた。これは，ごくわずかであるが，ディスプレイ上の色の違いがインクの濃淡の違いで表せたためである。そのため，PC-9801でプログラムを作成し，ハードコピーはFM16βを用いて出力した。この操作は，プログラム自体は変更することなしに，インクルードするグラフィック・ライブラリのファイル名だけを変えれば簡単に行える。FM用のグラフィック・ライブラリは，筆者が情報処理の授業で使っている「Pascal入門(東大出版会)」という本を参照されたい。

今回から，図の表示にCANONのLASER SHOTを用いることにする。レーザー・プリンタは精度が良く，最近価格が求めやすくなったため，使っている人も結構多いのではないだろうか。この場合もインクルード・ファイルの名前だ

◉図1は次ページ。

図2　多角形の頂点列

*ながしま　しのぶ　東京大学　教養学部　情報・図形科学教室　〒153　東京都目黒区駒場3-8-1

▼図1　グラフィックス・ライブラリ

```
  1: {  ***  ＰＣ-9801用 グラフィクス・ライブラリ  ***  }
  2: {  ファイル名  plot.lib  }
  3: {  各機能の説明
  4:    g_init                 : 初期化
  5:    g_cls                  : 画面消去
  6:    g_line(x1,y1,x2,y2)    : 線分の描画   (x1,y1)から(x2,y2)まで
  7:    g_color(i)             : 線分のカラーコード i=0～7
  8:    g_copy(i)              : ハードコピー i=1:文字,i=2:図形,i=3:両方
  9:
 10: type register = record
 11:         ax, bx, cx, dx, bp, si, di, ds, es, flags : integer;
 12:       end;
 13:       screen_type = record
 14:         mode, switch, active, display : byte;
 15:       end;
 16:       line_type = record
 17:         x1, y1, x2, y2 : integer;
 18:         color, boxtype, linestyle, dum1, dum2, dum3 : byte;
 19:         tile1, tile2 : integer;
 20:       end;
 21:
 22: var g_workbuffer : array[1..$10] of byte;
 23:     g_workarea   : array[1..$1400] of byte;
 24:     screen_arg : screen_type absolute g_workarea;
 25:     line_arg   : line_type absolute g_workarea;
 26:     g_reg : register;
 27:     g_worksegbase, g_workoffset : integer;
 28:
 29: procedure set_g_reg;
 30: begin
 31:   with g_reg do begin
 32:     bx := g_workoffset;
 33:     ds := g_worksegbase;
 34:   end;
 35: end;
 36:
 37: procedure g_init;                  {  初期化  }
 38:
 39:   procedure set_screen;
 40:   begin
 41:     with screen_arg do begin
 42:       mode := 3;
 43:       switch := 0;
 44:       active := 0;
 45:       display := 1;
 46:     end;
 47:     set_g_reg;
 48:     intr($A1,g_reg);
 49:   end;
 50:
 51:   procedure set_line_arg_unused;
 52:   begin
 53:     with line_arg do begin
 54:       color := 7;
 55:       boxtype := 0;
 56:       linestyle := 0;
 57:       dum1 := 0;
 58:       dum2 := 0;
 59:       dum3 := 0;
 60:       tile1 := 0;
 61:       tile2 := 0;
 62:     end;
 63:   end;
 64:
 65:   procedure set_intr_lio;
 66:   var i : byte; lio_seg : integer;
 67:   begin
 68:     lio_seg := $F990;
 69:     for i := 0 to 15 do begin
 70:       memw[0000 :($A0 + i)*4] := memw[lio_seg : 6 + i*4];
 71:       memw[0000 :($A0 + i)*4 + 2] := lio_seg;
 72:     end;
 73:     memw[0000 : $CE*4] := memw[lio_seg : 70];
 74:     memw[0000 : $CE*4 + 2] := lio_seg;
 75:     memw[0000 : $C5*4] := $082B;
 76:     memw[0000 : $C5*4 + 2] := $FD80;
 77:   end;
 78:
 79: begin
 80:   set_intr_lio;
 81:   set_line_arg_unused;
 82:   g_worksegbase := dseg + ofs(g_workarea) div 16;
 83:   g_workoffset  := ofs(g_workarea) mod 16;
 84:   set_g_reg;
 85:   intr($A0,g_reg);
 86:   set_screen;
 87: end;
 88:
 89: procedure g_cls;                  {  画面消去  }
 90: begin
 91:   set_g_reg;
 92:   intr($A5,g_reg);
 93: { write(#27'[>1h');  文字消去を行なう場合  }
 94: end;
 95:
 96: procedure g_line(gx1, gy1, gx2, gy2 : integer);  {  線分の描画  }
 97: begin
 98:   with line_arg do begin
 99:     x1 := gx1;
100:     y1 := gy1;
101:     x2 := gx2;
102:     y2 := gy2;
103:   end;
104:   set_g_reg;
105:   intr($A7,g_reg);
106: end;
107:
108: procedure g_color(col : integer);  {  線分の色指定  }
109: begin
110:   with line_arg do begin
111:     color := col;
112:   end;
113: end;
114:
115: procedure hardcopy(switch : integer);   {  ハードコピー  }
116:
117:   procedure get_image(px, py : integer; nx, ny : byte;
118:                offset_copy, segment_copy : integer);
119:   begin
120:     with g_reg do begin
121:       ax := px;
122:       bx := py;
123:       cx := nx + ny * 256;
124:       di := offset_copy;
125:       es := segment_copy;
126:     end;
127:     intr($CE,g_reg);
128:   end;
129:
130:   procedure linefeed(i : byte);
131:   begin
132:     if i > 9 then write(lst,#27'T',i:0)
133:     else          write(lst,#27'T0',i:0);
134:     writeln(lst);
135:   end;
136:
137:   procedure printline_vram(low : byte);
138:
139:     procedure printchar_vram(vram : integer);
140:     var half_width : boolean;
141:     begin
142:       if (vram and $FF00) = 0 then begin
143:         if vram < $20 then vram :=$20;
144:         write(lst,chr(lo(vram)));
145:       end
146:       else begin
147:         if (vram and $80) = 0 then begin
148:           half_width := (lo(vram) <= $A) and (lo(vram) >= $9);
149:           write(lst,#27'K');
150:           if half_width then write(lst,#27#1)
151:           else                write(lst,#27#3);
152:           write(lst,chr($20 + lo(vram)),chr(hi(vram)));
153:           if half_width then write(lst,#27#2)
154:           else                write(lst,#27#3);
155:           write(lst,#27'H');
156:         end;
157:       end;
158:     end;
159:
160:     var tvramjis : array[1..25,1..80] of integer absolute $A000:0
161:         col : byte;
162:     begin
163:       for col := 1 to 80 do begin
164:         printchar_vram(tvramjis[low, col]);
165:       end;
166:     end;
167:
168:   var x, y, i, j : integer;
169:       gdataline : string[160];
170:       gdatac : array[0..160] of char absolute gdataline;
171:       gdata : array[0..160] of byte absolute gdatac;
172:       zero : char;
173:   begin
174:     if (switch <= 0) or (switch >= 4) then exit;
175:     write(#27,')3');
176:     zero:=chr(0);
177:     for j:=0 to 50 do begin
178:       write(lst,#27,'D',#27,'>');
179:       if switch >= 2 then begin
180:         for i:=0 to 3 do begin
181:           get_image(i*160,j*8,160,8,ofs(gdata[1]),seg(gdata));
182:           gdata[0]:=160;
183:           write(lst,#27,'S0160');
184:           write(lst,gdataline);
185:         end;
186:       end;
187:       if odd(switch) and (not odd(j)) and (j < 50) then begin
188:         linefeed(8);
189:         printline_vram(j div 2 + 1);
190:         write(lst,#$d);
191:         linefeed(8);
192:       end
193:       else linefeed(16);
194:     end;
195:     write(lst,#27,'A',#27,']');
196:     write(#27,')0');
197:   end;
```

◉図 2 は 171 ページ。

けを変えればよいようになっている。レーザー・プリンタへの図の出力については，次回説明する。

コンパイラは引き続き Turbo PASCAL Ver.3 を基本的に用いる。現在，Ver.4 や Ver.5 も発売中あるいは発売予定と聞いているが，途中で変更すると混乱を招くおそれがあるので，基本的には Ver.3 を用いる。もし，Ver.4 で図 1 のライブラリを使用する場合は，先頭に

uses dos, printer;

が必要で，さらに 26 行目の register を registers に，66 行目の integer を word に変えれば一応使用することができる。ただし，PC-98 用の Ver.4 には立派なグラフィックスがついているので，そちらを利用した方がよいかもしれない。

図形処理におけるデータ構造

これまでいくつかのワイヤー・フレーム構造を扱ってきたが，もう少しプログラム言語の特徴を生かしたデータ構造を考えてみたい。はじめに最も簡単な多角形のデータ構造について考えてみる。

図 2 のように，多角形は頂点列として表せる。最も簡単なのは次のようなものである。

頂点数 $= n$

座標値

x_1, y_1

x_2, y_2

x_3, y_3

…

x_n, y_n

線形変換など幾何的な数値計算で行える形状変形は容易だが，多角形の頂点の追加や多角形の切断による頂点削除などのデータの修正に手間がかかる。

頂点・稜線を別々のデータとして扱うと次のようになる。

頂点数 $= n$	稜線数 $= n$
座標値	始点・終点
$v_1=(x_1, y_1)$	$v_1\rightarrow v_2$
$v_2=(x_2, y_2)$	$v_2\rightarrow v_3$
$v_3=(x_3, y_3)$	$v_3\rightarrow v_4$
…	…

```
 1: program koch1;          {  *** 多角形の変形 ***  }
 2:
 3: {$I plot.lib}           {  グラフィクス・ライブラリ  }
 4:
 5: type pointer = ^cell;
 6:      cell = record
 7:        ip : pointer;
 8:        x, y : real;
 9:      end;
10: var h : real;
11:     root, tmp, next : pointer;
12:
13: procedure transform;      {  多角形の変形  }
14: var k : integer;
15:     tx, ty : real;
16:     tmp2 : pointer;
17: begin
18:   tmp := root;
19: { h := -h; }
20:   repeat
21: {   h := -h; }
22:     next := tmp^.ip;
23:     tx := next^.x - tmp^.x;
24:     ty := next^.y - tmp^.y;
25:     new(tmp2);                  {  新しい頂点  }
26:     tmp2^.x := (tmp^.x + next^.x)/2 - h*ty;
27:     tmp2^.y := (tmp^.y + next^.y)/2 + h*tx;
28:     tmp^.ip := tmp2;
29:     tmp2^.ip := next;
30:     tmp := next;
31:   until next^.ip = nil;
32: end;
33:
34: procedure display;          {  多角形の表示  }
35: var x, y, xo, yo : integer;
36: begin
37:   g_cls;
38:   tmp := root;
39:   repeat
40:     x := round( tmp^.x + 320);
41:     y := round(-tmp^.y + 200);
42:     if tmp <> root then g_line(xo,yo,x,y);
43:     xo := x;
44:     yo := y;
45:     tmp := tmp^.ip;
46:   until tmp = nil;
47: end;
48:
49: const np = 4;            {  メインプログラム  }
50:       rect : array[1..np,1..2] of integer =
51:         ((-1,1),(-1,-1),(1,-1),(1,1));
52: var i : integer;
53:     r : real;
54:     ans : char;
55: begin
56:   g_init;
57:   g_cls;
58:   g_color(5);
59:   write('r h ? ');
60:   readln(r,h);
61:   new(root);              {  正方形の生成  }
62:   tmp := root;
63:   for i:=1 to np do begin       {  座標値  }
64:     tmp^.x := r*rect[i,1];
65:     tmp^.y := r*rect[i,2];
66:     new(tmp^.ip);
67:     tmp := tmp^.ip;
68:   end;
69:   tmp^.x := root^.x;
70:   tmp^.y := root^.y;
71:   tmp^.ip := nil;
72:   repeat
73:     display;
74:     write('終了(y) ? ');
75:     readln(ans);
76:     if (ans = 'y') or (ans = 'Y') then exit;
77:     transform;
78:   until false;
79: end.
```

図 4　多角形変形プログラム

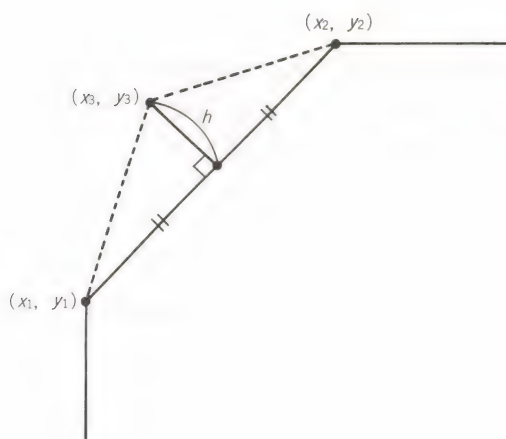
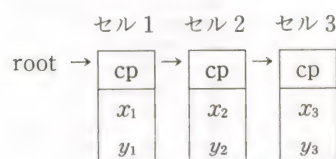



図3　新しい頂点の追加

このデータ構造の場合，頂点間に新しい頂点を追加することなど，上のデータ構造より簡単に行えることがわかる。

Pascalの動的データは必要に応じてメモリを使用していくもので，メモリが有効に使える。多角形を表現する場合は，次のように頂点データをポインタでつなげていくものである。

それぞれの要素はセルとよばれ，cpはセルへのポインタを表す。頂点の追加は新しいセルを生成してポインタを付け替えることにより，簡単に行える。

このような動的データ構造を利用して多角形や多面体のデータ構造を定義することができるが，データの部分的削除ができないのが大きな問題点である。しかし，この問題を改善したプログラムを組むこともできるので，ここでは動的データの練習として多角形の生成と変形を行ってみる。

多角形の変形

Pascalの動的データを用いて多角形を生成し，頂点を追加する処理を行ってみる。適当な例題が見つからないので，ここでは，**図3**のように図形を構成する線分をある規則により2個の線分に分け，これらの各線分に対して同じ規則を次々に適用していくことにより幾何学模様的な図形を生成していくことにする。

はじめに，(x_1, y_1)から(x_2, y_2)へ向かう線分があるとすると，次のようなデータができている。

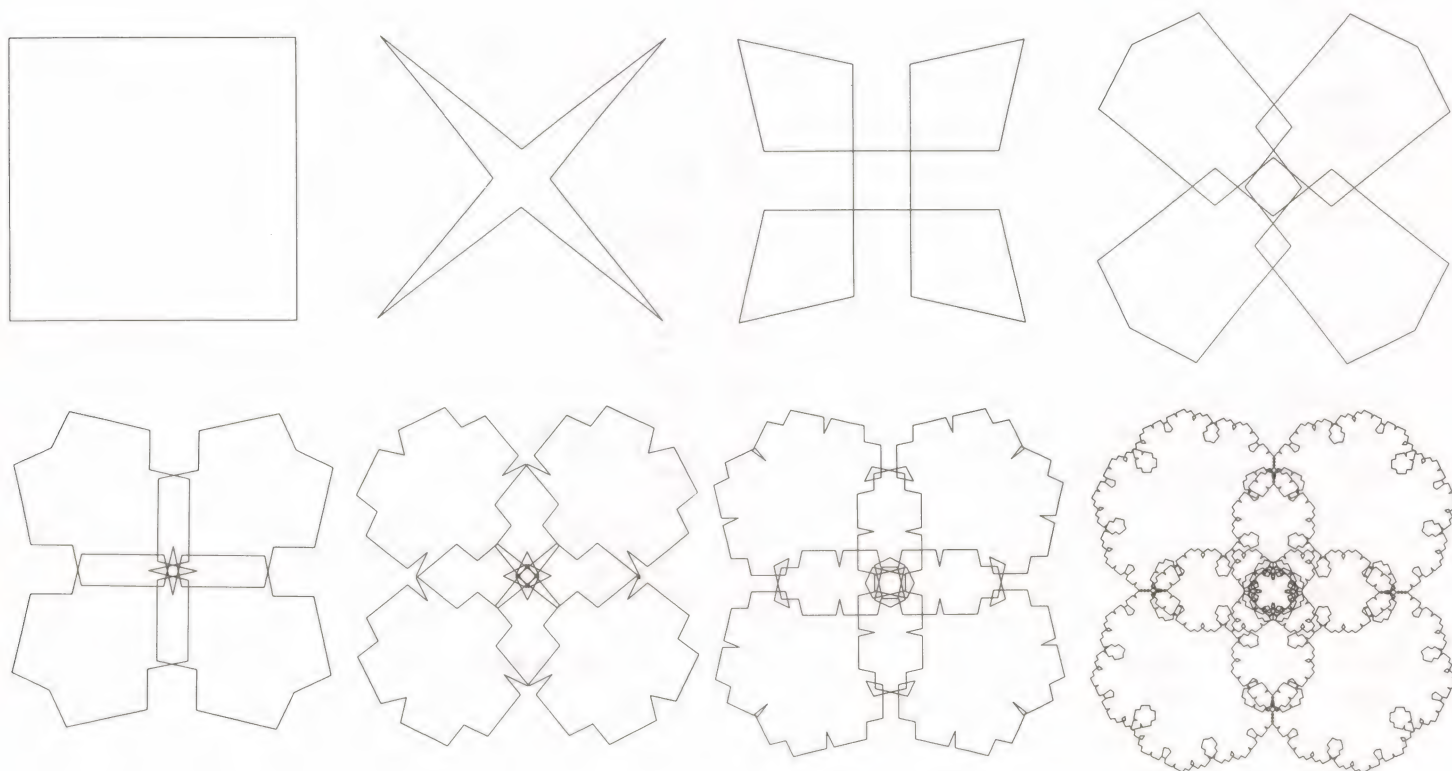



図5　プログラムの表示例1

◉図 4 は前ページ，図 7 は次ページ。

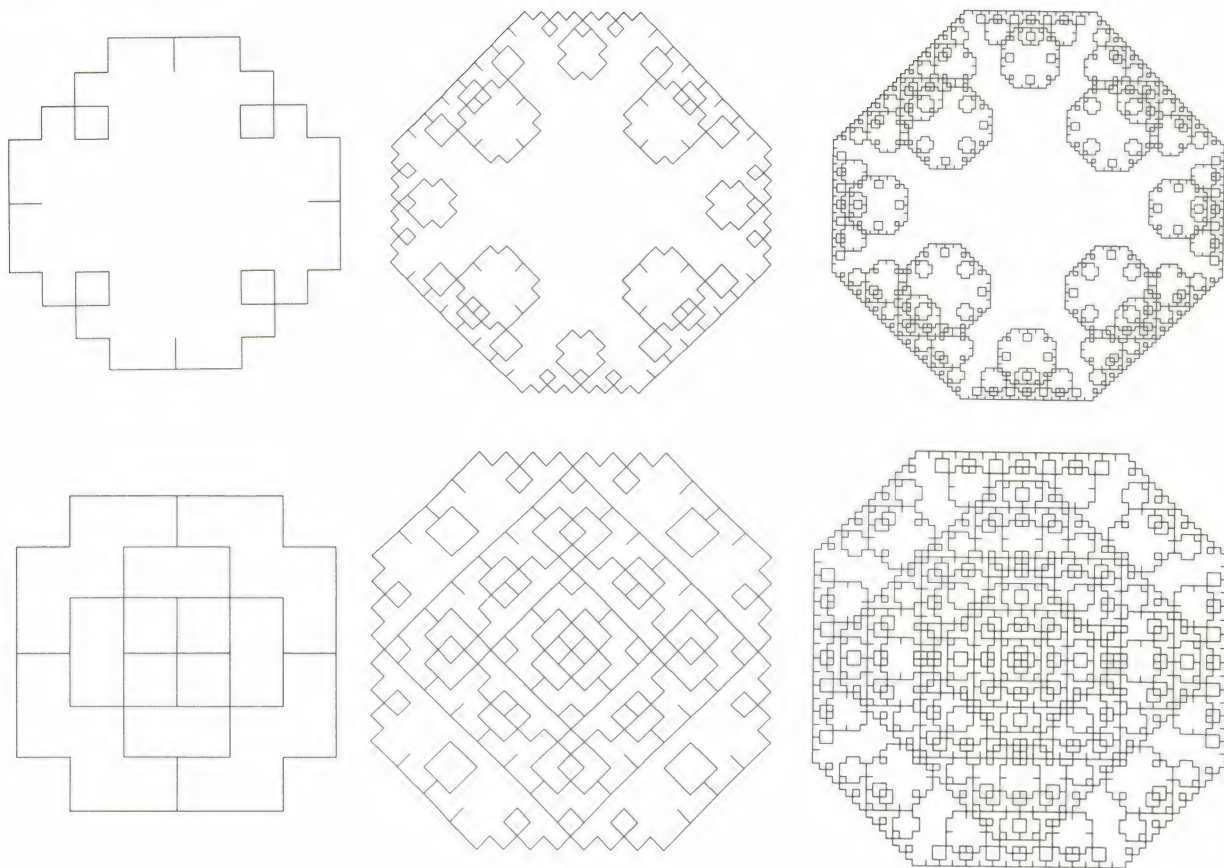
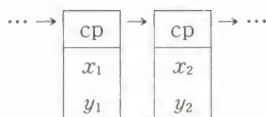
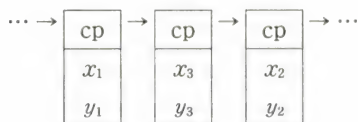



図 6　プログラムの表示例 2

この線分の途中に点 (x_3, y_3) を追加すると，次のようなデータになる。

この変更はポインタを付け替えることにより行える。

このような規則により正方形を変形して表示するプログラムを，**図 4** に示す。このプログラムは多角形を変形させる手続き，表示手続き，メインプログラムからなる。

メインプログラムでは，最初の正方形を生成し，変形と表示を繰り返す。頂点のデータの座標系は横方向が-320から$+320$，縦方向が-200から$+200$になっている。入力する r, h は正方形の 1 辺の長さ（の半分）と図 3 における h の値である。h はもとの線分の長さを 1 に正規化したときの値である。

手続き display では root から順に，つまり第 1 番目の頂点から順に頂点列をたどり，x，y 座標を読んで線分を描画していく。nil は End of data を表している。

手続き transform は，図 3 の規則に従って多角形の線分の中間に新しい頂点を生成する。tmp が i 番目の セル，next が $i+1$ 番目のセル，tmp2 が新たに生成されたセルを表している。tmp2 の x，y 座標を計算し，tmp のポインタが tmp2 を，tmp2 のポインタが next を指すようにすれば，線分の中間に頂点が生成される。この手続きが呼び出されるごとに，4 角形が 8 角形，16 角形，32 角形，…と変形を繰り返す。

このプログラムの表示例を**図 5** に示す。これは $r=100$，$h=0.4$ のときの変形していく表示である。r の値は最初の正方形の大きさ，つまり図の拡大率を表している。h を変えることにより，形がいろいろ変化する。

図 6 は上の段が $r=60$，$h=-0.5$，下の段が $r=80$，$h=0.5$ のときの表示で，途中の表示は大部分省略してある。線分どうしが重なり合って多角形ともいえないが，右側の表示は上下とも 4,096 角形である。**図 7** は $r=120$，$h=0.25$ と $r=25$，$h=1$ のときの表示である。このように h の値を変えることにより，さまざまな形の図形が得られる。さらに，図 4 の 19 行目または 21 行目のどちらかのコメントを外すことにより，いろいろなフラクタル図形が表示できる。

練習問題の解答

[12-1]　Pascal　の動的データを用いてワイヤー構造を作るには？

本文を参照のこと。

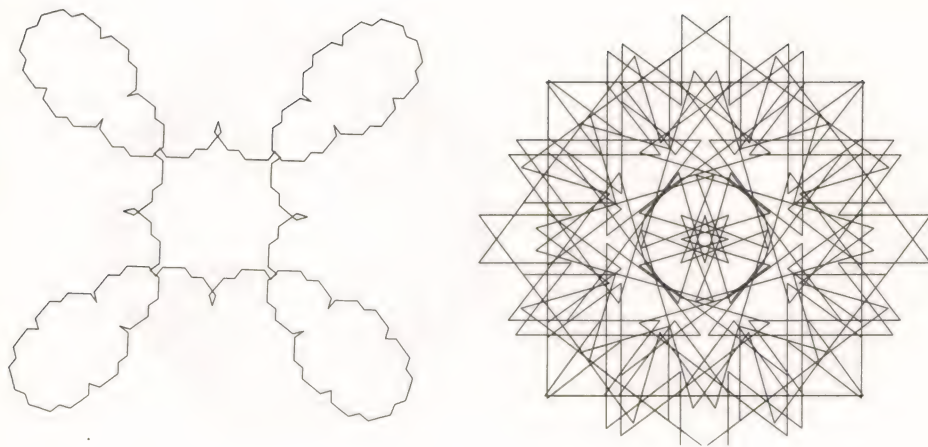



図7　プログラムの表示例3

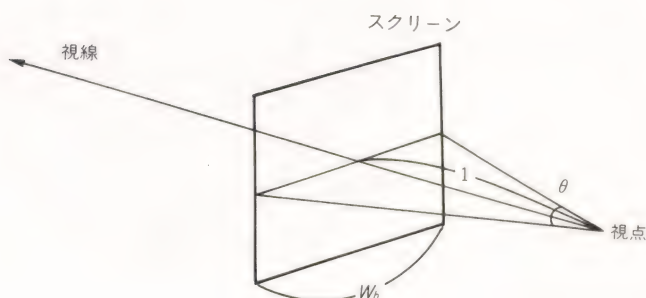



図8　視点とスクリーン

[12-2]　w_h の値と視野角度との関係は？

前回設定した視点とスクリーンとの位置関係では，視点とスクリーンとの距離は1であるから，水平方向の視野角度を θ とすると，**図8**のように

$\tan(\theta/2) = w_h/2$

$\theta = 2\tan^{-1}(w_h/2)$

である。例えば，

$w_h = 2$　では　$\theta = 90°$

$w_h = 1$　では　$\theta \fallingdotseq 53°$

$w_h = 0.5$　では　$\theta \fallingdotseq 28°$

になる。

[12-3]　3次元クリッピングを行うためにはどのようにすればよいか？

前回，少し説明したように，視点とスクリーンの4隅の点を結ぶ4角錐の内部が表示すべき領域である。

[12-4]　視線方向を左右・上下に回転するパーン・チルトを行うには？

視点を固定し，視心（注視点）を回転すればよい。頂点の位置を $e(e_x, e_y, e_z)$，視心の位置を $o'(o_x', o_y', o_z')$ とすると，θ，ϕ の正弦，余弦の値は，

$\sin\theta = (e_y - o_y')/a$

$\cos\theta = (e_x - o_x')/a$

$\sin\phi = (e_z - o_z')/b$

$\cos\phi = a/b$

$a = \sqrt{(e_x - o_x')^2 + (e_y - o_y')^2}$

$b = \sqrt{(e_x - o_x')^2 + (e_y - o_y')^2 + (e_z - o_z')^2}$

から計算できる。パーン・チルトを行わないときは，θ，ϕの値そのものは必要なく，θ，ϕ の正弦，余弦から変換行列を計算し，座標系の変換を行う。

$$T = \begin{bmatrix} \cos\phi\cos\theta & \cos\phi\sin\theta & \sin\phi \\ -\sin\theta & -\cos\theta & 0 \\ -\sin\phi\cos\theta & -\sin\phi\sin\theta & \cos\phi \end{bmatrix}$$

$$\begin{bmatrix} d \\ h \\ v \end{bmatrix} = T \begin{bmatrix} x - e_x \\ y - e_y \\ z - e_z \end{bmatrix}$$

パーン・チルトを行うには，T を z 軸・y 軸まわりに回転させればよい。例えば，$\Delta\theta$ の分パーンを行うには，

$$T' = \begin{bmatrix} \cos\Delta\theta & \sin\Delta\theta & 0 \\ -\sin\Delta\theta & \cos\Delta\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} T$$

とすればよい。

もう一つの方法として，逆三角関数から θ，ϕ の値を計算し，

$\theta_{new} = \theta + \Delta\theta$

$\phi_{new} = \phi + \Delta\phi$

として T を再計算する方法も考えられる。

これらのパーン・チルトを行うことにより，対象物から視線がそれることになる。CG のアニメーションでは視心を固定し，つまり画面に対象物を表示したまま視点を回転移動する表示方法をよくみる。

[12-5]　魚眼レンズを用いたように立体を表示するには？

スクリーンとして球面や円柱面を使うことが考えられる。魚眼レンズのように超広角になる。しかし，コンピュータのディスプレイは平面直交座標なので，球面から平面への投影変換が必要になる。この投影変換は地図を作成する際に用いられる。

[12-3][12-4][12-5]の問題は，機会があれば今後取り上げようと思っている。

練習問題

[13-1]　多角形の線分を複数個の線分に置き換える一般的なコッホ曲線の生成法は？　それを用いて雪の結晶のような雪片曲線を生成せよ。

[13-2]　乱数を用いて多角形を変形させよ。

[13-3]　一方向でなく，双方向参照できるデータ構造を考えよ。

[13-4]　多角形の頂点は最大いくつ生成できるか。

[13-5]　今回のプログラムを FORTRAN で実現するには？

カリブレーション機能付き高解像度カラー・モニター"CALIBRATOR"

東陽テクニカ

"CALIBRATOR"は，ベルギーのBARCO社が永年のカラー・モニター技術を駆使して開発したものです。最大の特徴は，今まで考えられなかった色の再現性を，モニター側でできるインテリジェント機能をモニターに付加したことです。

第1に，自動オフセット・レベル補正システムによりBlackレベルを常時補正し画像を安定させます。第2に，経時変化によるフォスファー変化を定期的に補正します。第3に，環境照明に対してコントラストを自動的に変化させ画面の反射を一定にします。この自動補正は，"CALIBRATOR"にCPUを内蔵し，AKB(自動オフセット・レベル補正)システムのソフトウエア化により実現されたものです。

色温度，コントラスト，ブライトネスのゲイン，画面のジオメトリはすべてパラメータ，つまりデジタル(数値)により制御できます。これらの数値データはKEYPADあるいはPCによっても簡単に制御できますので，そのデータの保存，持ち運び，さらにデータの再現が可能となります。

CALIBRATORとPCまたはKEYPAD間のインタフェースはRS-232Cです(RS-232Cはパラメータのみの通信で，ホスト・コンピュータとの通信はRGB信号です)。また，カラーの色差性能は，フォスファー座標(xyCIE 1931)に準じ，カリブレーションが機能しているため，カラーの標準機ともいえるモニターになり得ます。CRTは最新型のDAF(非集束型)電子銃を採用しているため，コンバージェンス比を0.1～0.35と最小限に抑えており，画面の隅々まで高画質が得られます。

映像機能は，20インチ，0.31ドットピッチ，解像度1,280×1,024，中心周波数48～66 kHz対応であり，入力信号はR, G, B信号でシンクオングリーン，コンポジット，セパレートのいずれも使用可能で，ループスルーBNC型を採用していますのであらゆるシステムに問題なく接続できます。

このような性能をもつ"CALIBRATOR"は，すでに欧米諸国で1,000台以上の販売実績をもち，日本国内でも印刷および写真業界のように色に対する要求が厳しい分野でかなりの需要が見込まれています。

問合せ先

㈱東陽テクニカ　貿易部

担当：山口，菅沢

☎03(245)0306　FAX 03(271)4757

RISC CPU「R3000」をベースにした低価格普及版「SUPER³(スーパーキューブ)1500」販売開始

日本電算機

日本電算機(石井孝利 代表取締役)は，米国MIPS社製RISC CPU「R 3000」をベースとしたグラフィック・スーパーコンピュータ「SUPER³1500」の販売と出荷を3月13日より開始しました。

日本国内で現在，供給可能なMIPS社製RISC CPU「R 3000」をベースにしたワークステーションでは，弊社製「SUPER³ 2000」とこの「SUPER³1500」だけです。

ワークステーション・エンジンとして25 MIPS，7 MFLOPS，グラフィック・エンジンとして高速13万ショート・ベクトル／秒に加えて，解像度は水平1,280ドット×垂直1,024ドット，最大256色同時表示可能という超高性能グラフィック・スーパーコンピュータです。OSにはUNIX System Vにバークレー・バージョン(4.3 BSD)を併合した新バージョンを搭載しています。

SUPER³ 1500

性能・機能の高いグラフィック・スーパーコンピュータの需要はこのところ急激に高まっていますが，弊社ではこうしたユーザーの動向に対応するために，今回の新製品を市場に投入することにしたものです。

新シリーズ「SUPER³」の詳細

- シリーズ名
  グラフィック スーパー コンピューター「SUPER³(スーパーキューブ)」
- 製品名
  「SUPER³1500」
- ワークステーション部の性能
  RISCアーキテクチャR3000/R3010，整数演算25MIPS,浮動小数点演算7 MFLOPS，主記憶容量16 Mバイト，ハードディスク容量171 Mバイト
- グラフィック部の性能
  13万ショート・ベクトル／秒，描画領域：水平1,280ドット×垂直1,024ドット，最大256色同時表示，フレーム構成：最大10プレーン(グラフィック・プレーン8，ワークプレーン1，キャラクタープレーン1プレーン)
- 周辺機器
  光ディスク，MT，デジタイザ，マウス，イメージスキャナ，ハードコピー各種，フィルムレコーダ，プロッタ
- オペレーション・システム
  UNIX System Vと4.3 BSDが併合した新バージョン
- 言語
  C，FORTRAN，LISP，Pascal，COBOL，Ada，PL/I
- ネットワーク
  Ethernet，TCP/IP，NFS，X-Window，DECnet
- インタフェース
  RS-232C(標準)，GP-IB，DRV 11 WA，SCSI，SCSI 2
- ソフトウエア
  JGL(グラフィックス・アンド・イメージング)，GKS
- データーベース・マネジメントシステム
  UNIFY，INFORMIX，INGRES
- 価格
  890万円より
- 発売時期
  1989年3月13日より
- 初年度販売予定台数
  国内200台，国外100台

NEWS SCAN

ICカード内蔵のドット・フォント6種を発表

文字図形センターは，ICカードを媒体とした文字フォントの販売を目的として昨年，設立された会社だが，このほどICカードを使ったレーザービーム・プリンタ(LBP)などに出力できるドット・フォント6書体を発表した。これは，三菱樹脂が開発したドライブに差し込んで使用する。

今回発表されたフォントは，明朝体，角ゴシック体，丸ゴシック体，教科書体，楷書体，行書体である。それぞれ48×48ドットのドット・フォントで，タイプバンク，印刷機械貿易，岩田母型，日本タイプライター，フォントランドにより書体が制作されたものである。

文字図形センターは，高品位文字フォントを文字メーカー(書体制作会社)から購入し，ICカードに収納して販売する。プリンタ，ワードプロセッサ，DTPソフトウエアなどへの対応を準備中である。

日本工学院専門学校がアポロ製WS280台を導入

分散型ワークステーションのメーカーである日本アポロコンピュータは，日本工学院専門学校から176台のワークステーションを新規受注し，これまでに導入された73台と合わせて249台が設置されることになった。学校関係ではこれまでにない大規模な導入例となる。新規受注総額は約4億円で，3月末までに納入された。

受注したワークステーションの中心機種は，4MIPSのデスクトップ・ワークステーションDN3500が114台で，さらにエントリ・レベルのワークステーションDN3000が55台導入される。

日本工学院専門学校では，選定の理由として，アポロの分散型アーキテクチャを評価したこと，トークンリング・ネットワークの応答性の良さ，UNIX System V，4.3BSD，AEGIS，MS-DOSが同時並行動作することなどをあげている。

同校では，アポロのワークステーションを大規模なCADシステムとしてネットワークの中に組み込み，UNIX，コンピュータ言語，CADなどの教育に利用していく予定である。

日本工学院専門学校からの大量受注の中心機種
4MIPSのデスクトップ・ワークステーション「シリーズ3500」

アマチュアCGコンテスト入賞作品決定

プロジェクトチームDo-GA主催による「第1回 アマチュアCGアニメーション コンテスト」の入賞作品が決定した。最優秀作品賞は該当なしとなったが，優秀作品賞に京都大学マイコンクラブ「つの虫君の大冒険」，梅沢順「PCGA」，鳥取大学電子計算研究会「ART-2」が決定した。

この他に奨励賞として大阪府立大学RANDOM「The Fireworks」，大阪大学コンピュータクラブ「Mの喜劇」がそれぞれ受賞し，特別賞がCG連合，X68000(シャープ)に対して与えられた。

日本放送協会の為ケ谷秀一，日本電子専門学校の野地朱真，本誌編集長の河内隆幸ほか7名が審査にあたった。

武内　重親氏

アップル コンピュータ ジャパンの代表取締役社長に武内重親氏が就任

アップル コンピュータ ジャパンは，3月6日より代表取締役社長に武内重親氏が就任したと発表した。同社は昨年10月に初めての日本人社長として堀昭一氏を起用し，日本市場における販売促進，顧客サービスやサポート強化，R&Dセンターの開設などを行ってきた。

武内氏は前東芝ヨーロッパ上級副社長で，同社の経営戦略，市場／製品戦略を担当していた。アップル コンピュータ ジャパンではMacintoshをはじめとする製品の販売，広報宣伝活動を強化していく方針である。今年9月までに従業員数を倍増させ，1992年から93年にかけての売上高を5億ドルにもっていく計画である。

CD-ROMを使った静止画ハイビジョンギャラリー・システムが完成

NHKエンジニアリングサービス，NHKエンタープライズ，大日本印刷，池上通信機，日本ビクターの5社は共同で，ハイビジョン・グラフィックス(静止画)による「ハイビジョンギャラリー」の実用化システムを完成させた。その第1号として4月4日に世界初のハイビジョンギャラリーが岐阜県美術館に開設された。

ハイビジョン・グラフィックスは，スチル写真を印刷用スキャナによってデジ

静止画ハイビジョンギャラリー・システム

塚田氏の作品

タル信号で取り込み画像処理を行った後，600 M バイトの容量をもつCD-ROMを利用したハイビジョン静止画ディスクに記憶するシステムである。1枚のディスクには数百点の作品を収録することができ，CD-ROMドライブを増設することにより，より大容量のシステムを構築することができる。マウスによる検索で，作家別・年代別に自由にアクセスでき，保存性も高い。また，PCM音声による解説も聞ける。多量の所蔵品をもつ美術館では特に有用であるとしている。画像は背面投射型の大型プロジェクタ(110インチ)で展示されるので，明るい環境でも鮮明な画像が得られる。

岐阜県美術館におけるハイビジョンギャラリーは面積約190 m²，固定席48名のシステムで，CD-ROMソフトウエアは約5分間のプログラムが20番組用意されている。所蔵作品1,900点のうち，現在約700点がデータベース化されており，作品名，制作者名，技法，素材，制作年代，題材など30項目からの検索が可能となっている。

第5回ヒューマン・インタフェース・シンポジウム

計測自動制御学会は，ヒューマン・インタフェースに関するさまざまな研究を発表するための発表講演会である「第5回ヒューマン・インタフェース・シンポジウム」を，10月25日から27日までの3日間にわたり開催する。映像利用技術，ハイパーメディア，メンタル・モデルといった新しいテーマに関する発表も歓迎している。

開催の概要は次の通り。

主催　計測自動制御学会
企画　ヒューマン・インタフェース部会
会期
講演会：10月25日(水)
シンポジウム：10月26日(木)~27日(金)
会場　アピカルイン京都(京都市左京区)
問合せ先　京都大学　工学部航空工学教室　井上研究室内　第5回ヒューマン・インタフェース・シンポジウム事務局
☎ 075(753)5795

スーパーコンピュータ研究所がスーパーコンピューティングのニューズレターを発刊

スーパーコンピュータの基礎研究機関であるスーパーコンピュータ研究所(ISR)は，スーパーコンピュータ・アーキテクチャやアプリケーション・ソフトウエア/ハードウエアの研究・開発などの記事を英語で紹介する隔月刊のニューズレター"Vector Register"を発刊，読者を募集している。

レターサイズ，20ページのもので，日本語要約が1ページ付いている。現在3号まで出されており，購読料は無料。

問合せ先　㈱リクルート　スーパーコンピュータ研究所
〒104　東京都中央区勝どき2-11　リクルート勝どきビル
☎ 03(536)9661

3次元CG イラストレーション個展

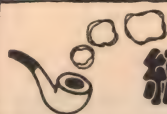
第1回PIXEL CGグランプリで優秀賞を獲得したCGアーチスト，フジカラーサービスCGセンターの塚田哲也氏の3次元CGイラストレーション作品展「ぼくはセールスマン」が，4月20日から25日まで東京・渋谷のアートワッズで行われる。

展示されるのは，ポジフィルムからインクジェット・プリンタ，ピクトログラフィでダイレクトプリントしたもの，ビデオプリンタから出力したもの，CG作成のための3次元原データ，計算中の絵など30~40点である。

会期　4月20日(木)~25日(火)
会場　アートワッズ　☎ 03(770)1929
問合せ先　フジカラーサービス　塚田哲也　☎ 03(567)8295

キャダム社が米国のパーソナルCADベンターを買収

米国キャダム社(本社：カリフォルニア州バーバンク)は，パーソナルコンピュータ・ベースのPCB設計用CAD/CAEソフトウエア「P-CAD」を開発・販売しているパーソナルCADシステムズ社(本社：カリフォルニア州サンノゼ)を買収した。キャダム社のこの買収によってP-CADのユーザーは，CADAMファミリーへのスムーズな統合が可能になるとしている。

P-CADは，PCB用CAD/CAEパッケージとしてはインストール・ベースで11,000件と，世界最大のユーザー数をもっている。CADAMは35,000端末，15万件のユーザーをかかえている。また，メインフレーム用のCADAM Interactive Design，ワークステーション・ベースのPROFESSIONAL CADAM，パーソナルコンピュータ用のMICRO CADAMの製品ラインをもつ。

編集後記

◘ネットワーク技術やワークステーション製品がマルチベンダーを推進し，ハードウエア・メーカー・オリエンティッドな考え方を修正しつつあります。これまでコンピュータ・システムを考えるとき，どこのコンピュータがいいのかということがまず議題になり，コンピュータ・メーカーが決まってしまうと，そのコンピュータに依存した話から抜け出すことができなくなり，何年かたって次のコンピュータにリプレイスする場合もデータベースやソフトウエア・プログラム，ホストに依存した端末や周辺機器の問題などがでてきて，別のコンピュータ・メーカーにするということは大変に勇気のいることでありました。

しかし，最近はそのような事情も少しずつ変わり始めました。ハードウエアについては，各部門ごとに最適なものを選択する（ソフトウエアについても同様なのですが），それをネットワークに乗せてデータの流通を図るようになりつつあります。同じプログラムがいくつかの主要なワークステーションに搭載されていることが普通になってきましたので，ユーザーとしても同時にいくつかのメーカーのワークステーションを購入することが習慣のようにさえなっています。ワークステーションは今後とも激しい競争が見込まれていますから，現在メジャーなワークステーションが今後ともそうあり続けるかどうかの保証はないわけです。

現在使用しているワークステーションを変更することになっても，アプリケーション・プログラムは今後とも同じものを使いたい，万が一アプリケーションの具合が悪くなったとしても，最低限度，そのデータファイルだけは使えなければいけません。というわけで，これまでのようにメーカーと心中するような環境は避けなければいけません。例えば，メインフレーム・メーカーもユーザーに適したコンピュータ・システムを提供するためにシステム・インテグレーション重視として，場合によっては他社のハードウエアでも良いものがあれば使うというように軟化してきています。

◉本誌89年3月号，表紙の解説のタイトルは「風雅」でした。また，4月号のCGグランプリ入賞者「長継重敏」は「長続重敏」氏の誤りでした。ここに訂正し，お詫び申し上げます。

次号のご案内

89年6月号(5月20日発売)

大特集　ワークステーションのすべて
◎ワークステーションとは何か
◎ワークステーションに使われているいろいろなCPU
◎話題のOS Mach
◎グラフィックス標準のいろいろ
◎WS各社のグラフィックス
●シリコングラフィックス　●ステラ
●タイタン　●サンマイクロシステムズ
●アポロ　●インターグラフ
●テクトロニクス　●日本電算機

特集　私のCG作法
●梅村高　●増尾隆幸　●川口吾妻
●佐藤篤司　●須藤牧人　●高根寿明
●塚田哲也　●長続重敏　●平野正史
(敬称略，順不同)

シリーズ　レンダリング・アプリケーション
●陶磁器，タイルにおけるCG

シリーズ　建築,土木のCADとCG
●ビル建築における耐震シミュレーション

その他の主な記事
◇テクスチャの表現
◇モデリングからレンダリングまでのCGの基礎
◇CGのための図学
◇デザイン・プロのための2次元CGテクニック

定期購読・別冊号のお申込み方法

4月1日より消費税の導入に伴い，PIXELおよび別冊号などに消費税3％が加算されますので，ご注意下さい。

PIXELの新しい定期購読料は消費税・送料込みで，
1年（12冊分）　11,840円
（本体11,500円　消費税340円）
半年間（6冊分）　5,970円
（本体5,800円　消費税170円）
になります。

PIXELは毎月1日発行(20日書店発売)で全国の書店で発売されますが，最寄りの書店にない場合は，直接購読をお勧めします。なお，特大号が含まれる場合も購読料は変わりません。新規に定期購読をお申し込みになられる方は，とじ込み葉書をご使用下さい。購読料のお支払いは後日，雑誌とともにお送りします郵便振替用紙にてお支払い下さい。

また，PIXELの1冊の定価（消費税込み）も変更になります。
定価1,010円
（本体981円＋税29円）
バックナンバーをお申込みの場合には送料が必要になります(送料96円～)。

定期購読継続のお申込み方法

本号で定期購読が切れた方は，巻末ハガキを利用して継続の手続きをして下さい。その際，必ず購読者番号をご記入下さい。
<申込み先：販売部>

広告のお申込み方法

本誌に広告掲載を検討されている方は，広告媒体資料をお送りいたします。
<申込み先：広告部>
お申込みはFAXでもお受けいたしております。
FAX：03(293)6164

PIXEL　89／5月号
1989年5月1日発行　No.80
編集・発行人　河内　隆幸
発行所　図形処理情報センター
〒101　東京都千代田区神田神保町1-64
神保町協和ビル6階　電話03(293)6161(代表)
発売元　オーク出版サービス　電話03(291)7031
印刷所　千代田平版社
定　価　1,010円(本体981円＋税29円，送料96円)
郵便振替　東京　5-50653

PIXEL 89/5月号

PIXEL 資料請求

ご入用な広告ページの資料請求番号を○でお囲み下さい

表2 表3 表4 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17
18 19 20 21 22 23 24 25 26 27 28 29 30 31

PIXEL アンケート

・本号でおもしろかった記事のタイトルを2つご記入下さい
1)
2)
・本号でおもしろくなかった記事のタイトルを2つご記入下さい
1)
2)
・本誌に対する要望, 感想, 批判など自由にお書き下さい

・本誌のお買い求め方法は　直接購読／書店
・当センターからのDMは　届いている／届いていないので送ってほしい／不要

PIXEL 定期購読・別冊号申込

請求書：要／不要

□本号で定期購読がきれましたので '89年6月号より　半年／1年間　継続を申し込みます／中止します　(購読者No.　　　　)
□新規に定期購読を '89年6月号より　半年／1年間　申し込みます
□バックナンバーを申し込みます

「PIXEL」
'89/1, 2, 3, 4
'88/1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12
'87/1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12

別冊
2号「アプリケーション」
4号「グラフィックスガイド」
5号「CAD/CAM/CAEの基礎」
CAD/CAMアルファNo.1／No.2

□別冊7号「CAD/CAM, CG年鑑89」を申し込みます。□別冊6号「CAD/CAM, CG総覧88」を申し込みます。

お送り先（葉書のおもて面と同じ場合は不要）
住所 〒
社名・部署
氏名　　　　電話

PIXEL 89/5月号

PIXEL 資料請求

ご入用な広告ページの資料請求番号を○でお囲み下さい

表2 表3 表4 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17
18 19 20 21 22 23 24 25 26 27 28 29 30 31

PIXEL アンケート

・本号でおもしろかった記事のタイトルを2つご記入下さい
1)
2)
・本号でおもしろくなかった記事のタイトルを2つご記入下さい
1)
2)
・本誌に対する要望, 感想, 批判など自由にお書き下さい

・本誌のお買い求め方法は　直接購読／書店
・当センターからのDMは　届いている／届いていないので送ってほしい／不要

PIXEL 定期購読・別冊号申込

請求書：要／不要

□本号で定期購読がきれましたので '89年6月号より　半年／1年間　継続を申し込みます／中止します　(購読者No.　　　　)
□新規に定期購読を '89年6月号より　半年／1年間　申し込みます
□バックナンバーを申し込みます

「PIXEL」
'89/1, 2, 3, 4
'88/1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12
'87/1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12

別冊
2号「アプリケーション」
4号「グラフィックスガイド」
5号「CAD/CAM/CAEの基礎」
CAD/CAMアルファNo.1／No.2

□別冊7号「CAD/CAM, CG年鑑89」を申し込みます。□別冊6号「CAD/CAM, CG総覧88」を申し込みます。

お送り先（葉書のおもて面と同じ場合は不要）
住所 〒
社名・部署
氏名　　　　電話

PIXEL 89/5月号

PIXEL 資料請求

ご入用な広告ページの資料請求番号を○でお囲み下さい

表2 表3 表4 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17
18 19 20 21 22 23 24 25 26 27 28 29 30 31

PIXEL アンケート

・本号でおもしろかった記事のタイトルを2つご記入下さい
1)
2)
・本号でおもしろくなかった記事のタイトルを2つご記入下さい
1)
2)
・本誌に対する要望, 感想, 批判など自由にお書き下さい

・本誌のお買い求め方法は　直接購読／書店
・当センターからのDMは　届いている／届いていないので送ってほしい／不要

PIXEL 定期購読・別冊号申込

請求書：要／不要

□本号で定期購読がきれましたので '89年6月号より　半年／1年間　継続を申し込みます／中止します　(購読者No.　　　　)
□新規に定期購読を '89年6月号より　半年／1年間　申し込みます
□バックナンバーを申し込みます

「PIXEL」
'89/1, 2, 3, 4
'88/1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12
'87/1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12

別冊
2号「アプリケーション」
4号「グラフィックスガイド」
5号「CAD/CAM/CAEの基礎」
CAD/CAMアルファNo.1／No.2

□別冊7号「CAD/CAM, CG年鑑89」を申し込みます。□別冊6号「CAD/CAM, CG総覧88」を申し込みます。

お送り先（葉書のおもて面と同じ場合は不要）
住所 〒
社名・部署
氏名　　　　電話

1989年5月1日発行/毎月1回1日発行/通巻80号
1984年5月2日第3種郵便物認可

図形処理情報センター 〒101 東京都千代田区神田神保町1-64 神保町協和ビル 電話 03(293)6161

定価 1,010円（本体981円＋税29円）（資料請求番号 表4）

雑誌 07651-5

PIXEL

コンピュータ・グラフィックスとCAD/CAM/CAE雑誌ピクセル

グラフィック・デザインにおけるCG

89
5

図形処理情報センター